



The Definitive Certified Scrum Product Owner® (CSPO) Workbook

Daniel James Gullo

CEC, CST

Copyright © 2012-2025

Daniel James Gullo
CEO, Principal

Apple Brook Consulting®
PO Box 369
Cochranville, PA 19330-0369

daniel.gullo@apple-brook.com

Disclaimer

Every effort possible has been made to provide proper attribution for the sources in this training handbook. Any missed attribution or misattribution is unintentional. Please contact us immediately, and we will revise accordingly.



Table of Contents

BUILD YOUR OWN SCRUM	4
SCRUM OVERVIEW	6
THE NEW NEW PRODUCT DEVELOPMENT GAME	7
WHAT DOES SCRUM STAND FOR?	8
SCRUM OVERVIEW	9
DEFINE EMPIRICISM AND LIST THE THREE EMPIRICAL SCRUM PILLARS	11
LIST THE FIVE SCRUM VALUES	12
SIGNIFICANT ORGANIZATIONAL CHANGE	13
PRODUCT OWNER CORE COMPETENCIES	14
CSPO-1.1 Discuss at least three types of organizational contexts	15
CSPO-1.6 Discuss at least two reasons why the Product Owner is a single person	16
CSPO-1.7 Discuss how and why the Product Owner maintains authority	16
CSPO-1.2 Use at least one technique to provide transparency to stakeholders on progress toward goals	22
CSPO-1.3 List at least three techniques to gather information or insights from stakeholders	26
CSPO-1.5 Identify at least two ways to overcome the challenges	27
GOAL SETTING AND PLANNING	28
CSPO-2.1 Explain the relationship between a product vision and a Product Goal	32
CSPO-2.2 Practice the creation of a Product Goal	34
CSPO-2.5 Describe how to plan a product release	39
CSPO-2.3 Creating a Sprint Goal with a Scrum Team	41
UNDERSTANDING CUSTOMERS AND USERS	42
CSPO-3.1 Describe how a Product Owner includes product discovery	43
CSPO-3.2 Illustrate at least one approach for segmenting customers and users	44
CSPO-3.3 Practice at least one technique to address conflicting customer (or user) needs	44
CSPO-3.4 Identify how at least three aspects of product discovery contribute to successful product outcomes	51
CSPO-3.5 List at least three approaches to connect Developers	52
WORKING WITH THE PRODUCT BACKLOG	53
CSPO-5.1 Describe the relationship between outcome and output	56
Asking ourselves "What is absolutely necessary?" in terms of documentation, process, meetings, etc. helps us to understand what we can get rid of in terms of wasteful overhead that doesn't add value to the customer / users.	56
CSPO-5.2 List at least three actions the Product Owner could take to maximize outcomes and impact while minimizing outputs	57
CSPO-5.3 Define at least three terms related to product economics	61
VALIDATING PRODUCT ASSUMPTIONS	77
CSPO-4.1 Describe how Scrum supports validating product assumptions	78
CSPO-4.2 Compare at least three approaches to validate product assumptions by their cost and quality of learning	79
CSPO-5.6 Create a Product Backlog that supports the achievement of a Product Goal	88
CSPO-5.7 Create at least one Product Backlog item that includes a description of desired outcome and value	88
CSPO-5.8 Practice at least one approach to refining the Product Backlog	88
AFTERWORD	89
WHAT'S NEXT?	93
APPENDIX	96
THE SCRUM GUIDE	97
BIBLIOGRAPHY	111

Build Your Own Scrum

ACTIVITY

The class will be divided up into teams.

Each team will have their own breakout room and Mural space to work on. The goal for your team is to create a visual, flow diagram that teaches and explains the Scrum framework according to what the Scrum Guide says.

You may use the provided elements in the Mural space or you can use your own elements which you borrow from the Internet and upload from your own machine(s).

You will have **30 minutes** for this exercise. At the end, you and your team should be prepared to present and explain your diagram.

Please ensure that the content you select is PG rated.



Scrum Overview

The New New Product Development Game

Authors

Hiroataka Takeuchi

Ikujiro Nonaka

Publication

Harvard Business Review, January 1986

Summary

This article was a key inspiration for Schwaber and Sutherland in their creation of Scrum

The typical waterfall approach to software development, is compared to a relay race, where there is little collaboration between the team members.

The sign-offs in the phases of traditional software development are akin to passing of the baton in a relay race.

Once each group has done its part, the success is no longer their “problem”.

Instead, following a more rugby related approach where the teams are working closely together and passing the “ball” back and forth to accomplish their goals is a more preferred approach.

Collaboration happens, and the teams are more self-organizing so that they can figure out whatever needs to be done to achieve success.

The article includes numerous case studies from development projects across various industries.

Several graphics describe what we now call “Rational Unified Process” with the phases of development overlapping instead of being discrete and gated with formal sign-offs.



Hiroataka Takeuchi



Ikujiro Nonaka

Source: <https://hbr.org/1986/01/the-new-new-product-development-game>

What Does SCRUM Stand For?

In reality, Scrum doesn't stand for anything. It's not an acronym at all. It refers to the formation in rugby that occurs when the ball is brought back into play. It's the coming together of teams in order to move forward.

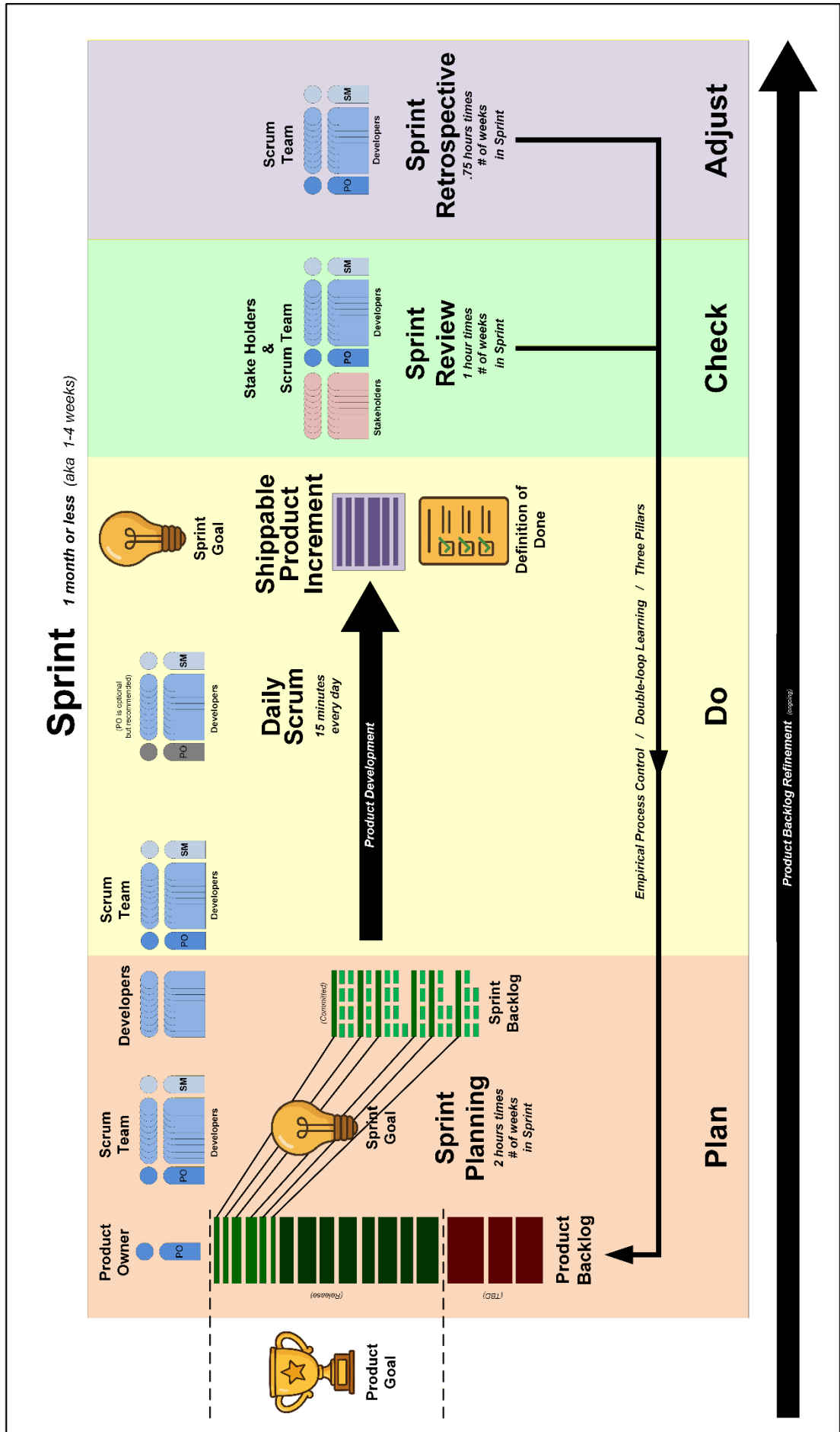


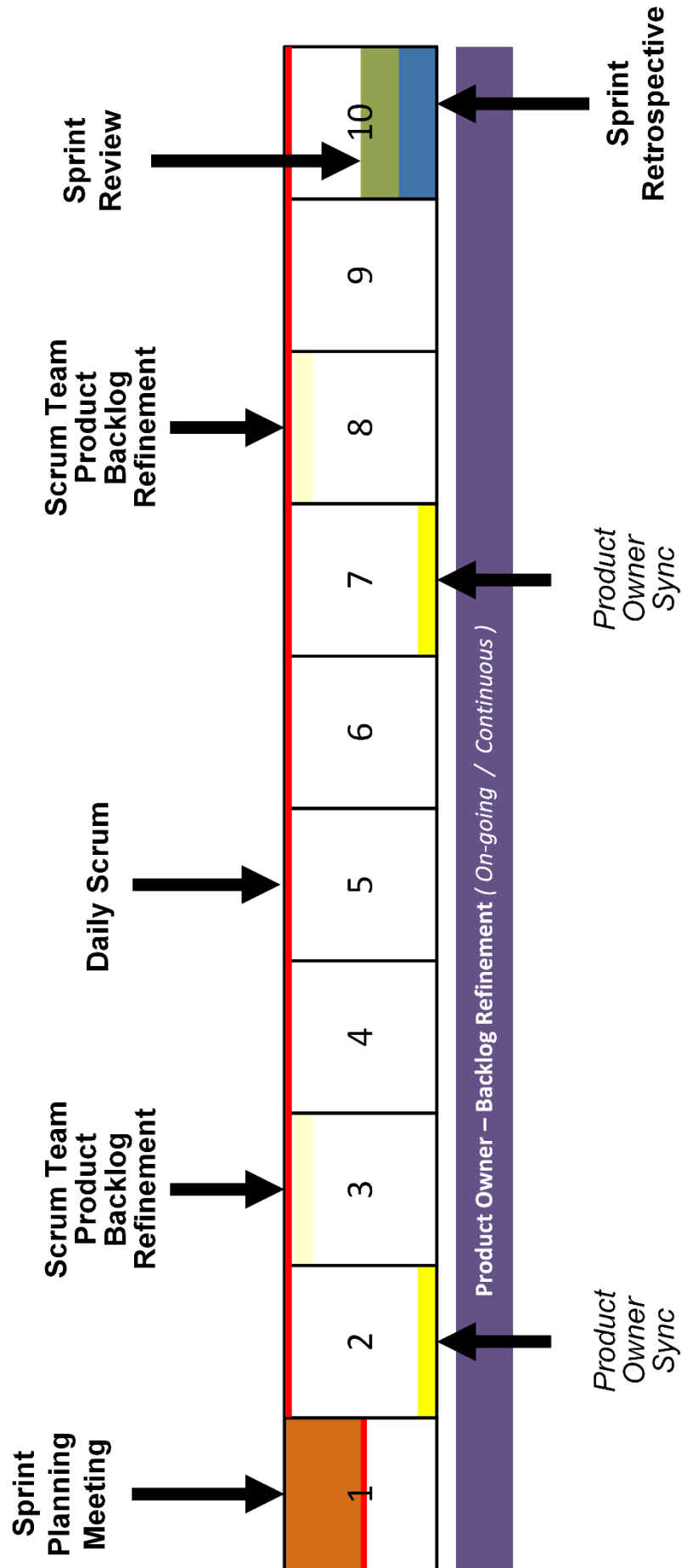
The emphasis is on a few baseline rules that outline a system for ensuring regular feedback loops to raise transparency.

Scrum is a Product Development framework and thus, says very little about Project Management specifically.

Scrum Framework

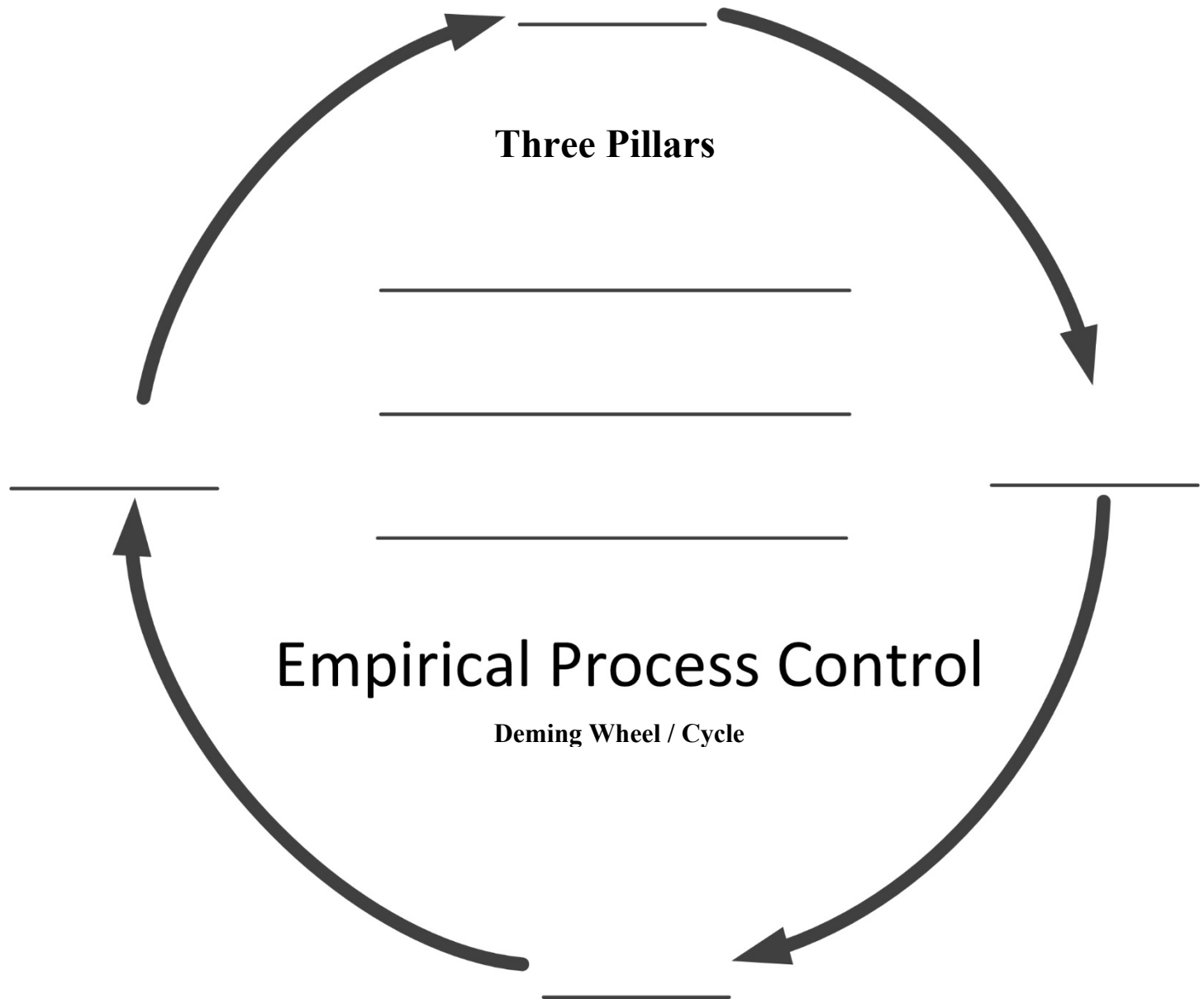
From The Scrum Guide by Ken Schwaber and Jeff Sutherland





Define empiricism and list the three empirical Scrum pillars

Scrum enables incremental and iterative development of products through various cycles or feedback loops which follow the pattern:



empirical - based on, concerned with, or verifiable by observation or experience rather than theory or pure logic

List the Five Scrum Values

Five Scrum Values

When the values of commitment, courage, focus, openness and respect are embodied and lived by the Scrum Team, the Scrum pillars of transparency, inspection, and adaptation come to life and build trust for everyone. The Scrum Team members learn and explore those values as they work with the Scrum events, roles and artifacts.

Successful use of Scrum depends on people becoming more proficient in living these five values.

People personally **COMMIT** to achieving
the goals of the Scrum Team.

The Scrum Team members have **COURAGE** to do the right thing and work on tough
problems.

Everyone **FOCUSES** on the work of the Sprint and the goals of the Scrum Team.

The Scrum Team and its stakeholders agree to be **OPEN** about all the work and the
challenges with performing the work.

Scrum Team members **RESPECT** each other
to be capable, independent people.

Working Agreements / Team Norms

Team operating model

*Who are we? What do we believe? How do we work?
What does it look like? Where are things documented?
What is our definition of done? What tools do we use?
Who are the backups? Etc.*

Sources:

*Ken Schwaber. Agile Software Development with Scrum (Upper Saddle River: Prentice Hall, 2002).
148-154.*

The Scrum Guide (2020)

Significant Organizational Change



Effective Scrum (Agile) is not simply sprinkled on top of the organization's current practices.

Current practices **MUST** change; which requires a change in mindset, priorities, and behavior.

Product Owner Core Competencies

CSPO-1.1 Discuss at least three types of organizational contexts that affect the approach to the Product Owner role

- **A Product Owner has complete ownership of target customer, problem, and solution**
- **A Product Owner owns the delivery of someone else's idea or initiative**
- **A Product Owner delivers a shared service to other Teams in the organization**
- **A Product Owner works on short-term projects that they own the outcome for**

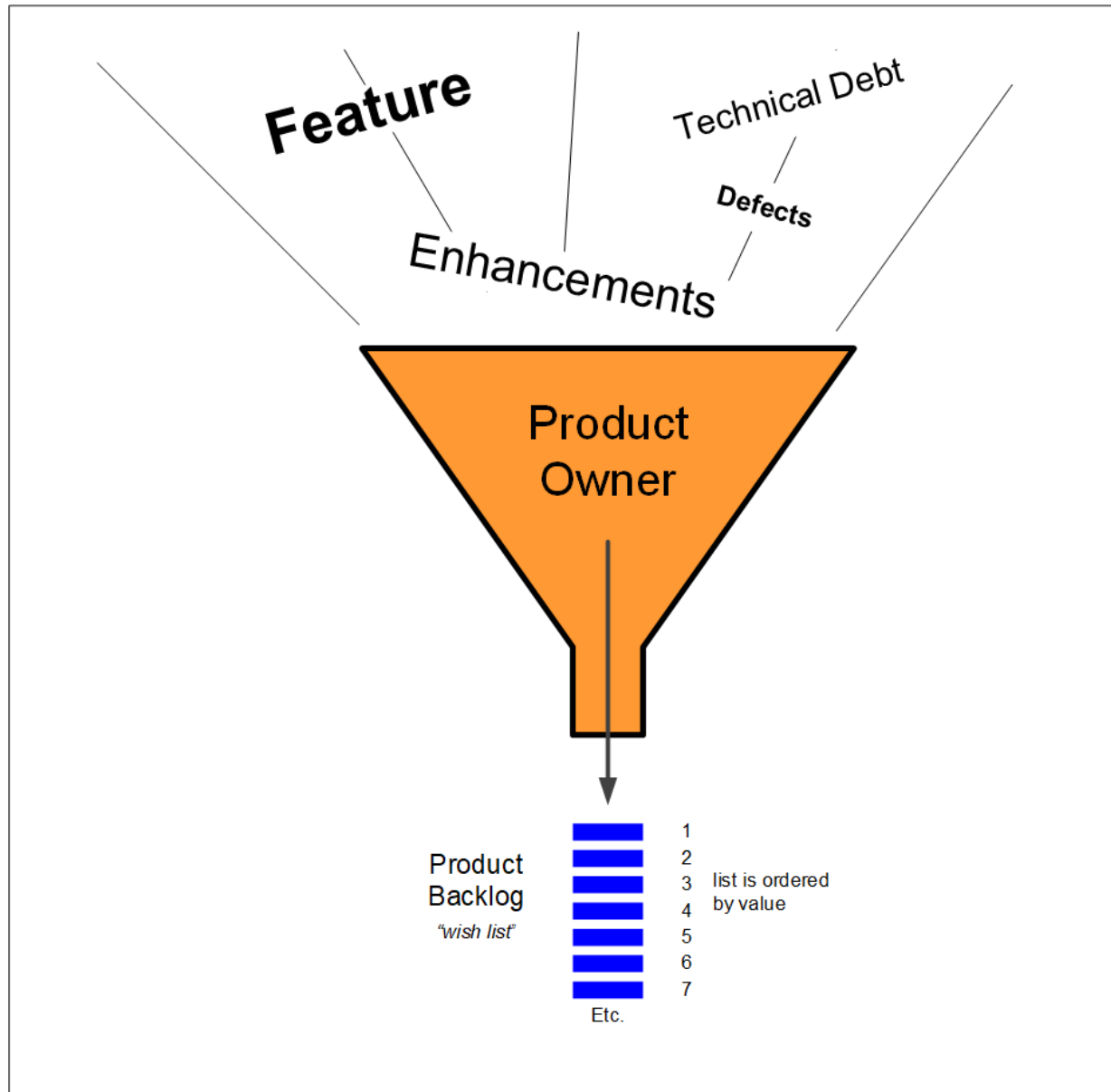
CSPO-1.6 Discuss at least two reasons why the Product Owner is a single person **and neither a group nor a committee**

CSPO-1.7 Discuss how and why the Product Owner maintains authority **over the Product Backlog while working collaboratively with Developers and stakeholders**

Accountabilities and Responsibilities

- Independent authority
- Empowered to make decisions
 - Backlog refinement
- Defining scope for the Developers
 - Deciding when to release
 - Ongoing visioning
- Budgeting / tracking finances
- Determining Release Dates
- Meets with customer regularly
 - Manages contracts
 - Deals with vendors
- Ensures product follows vision
- Balances customer (stakeholder) needs with organization needs

Product Owner Funnel



The Product Owner funnels ALL work requests intended for a Development Team into the ordered Product Backlog by making trade-offs between the various items.

RACI - Responsible, Accountable, Consulted, Informed

Responsible

Those who are doing the work and producing the output

Accountable

The ONE who ensures that the output results in valuable outcomes

Ultimately Accountable Ultimate Authority

Both are required.
Cannot have one without the other.

A Product Owner is an amalgam of the following traditional, waterfall style roles:

- **Product Manager**
- **Project Manager**
- **Business Analyst**
- **Marketing Analyst**
- **Customer Experience Designer**
- **Salesperson**
- **Business Process SME**

But **NOT** a Scrum Master!!

Too Busy

If you are doing everything in your power to make the product the best product it can be and maximizing value for the customer, you will not have any time to fulfill the Scrum Master Responsibilities.

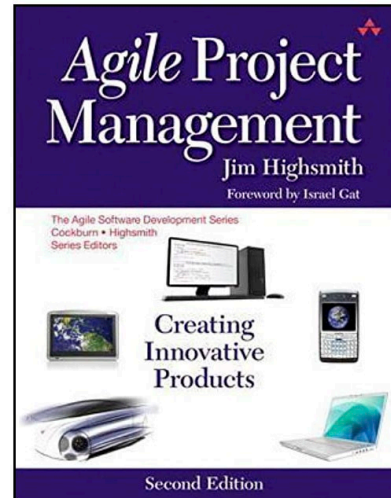
If the Scrum Master is doing everything in their power to maximize the effectiveness of the team and ensuring that all impediments are removed, etc. they will not have any time to fulfill the Product Owner responsibilities.

Conflict of Interest

Q: If the PO and Developers argue about something, who is going to resolve the conflict?

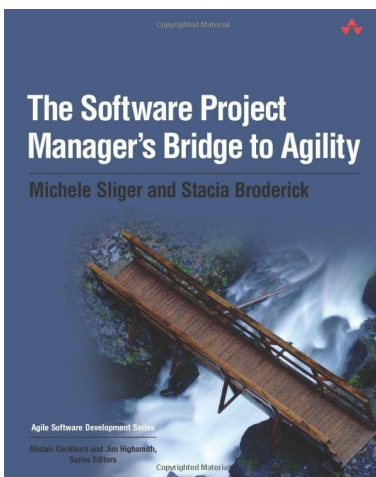
A: The Scrum Master

Consider the responsibilities of the
Traditional PM vs. Product Owner



“Process is not a substitute for skill.”

-Jim Highsmith



The Software Project Manager's Bridge to Agility
Michele Sliger and Stacia Broderick

CSPO-1.2 Use at least one technique to provide transparency to stakeholders on progress toward goals

- Cumulative Value chart
Using RICE, WSJF, WDM, Kano, etc.

- Roadmaps

- Sprint reviews

- Information Radiators
E.g. Product Strategy Canvas

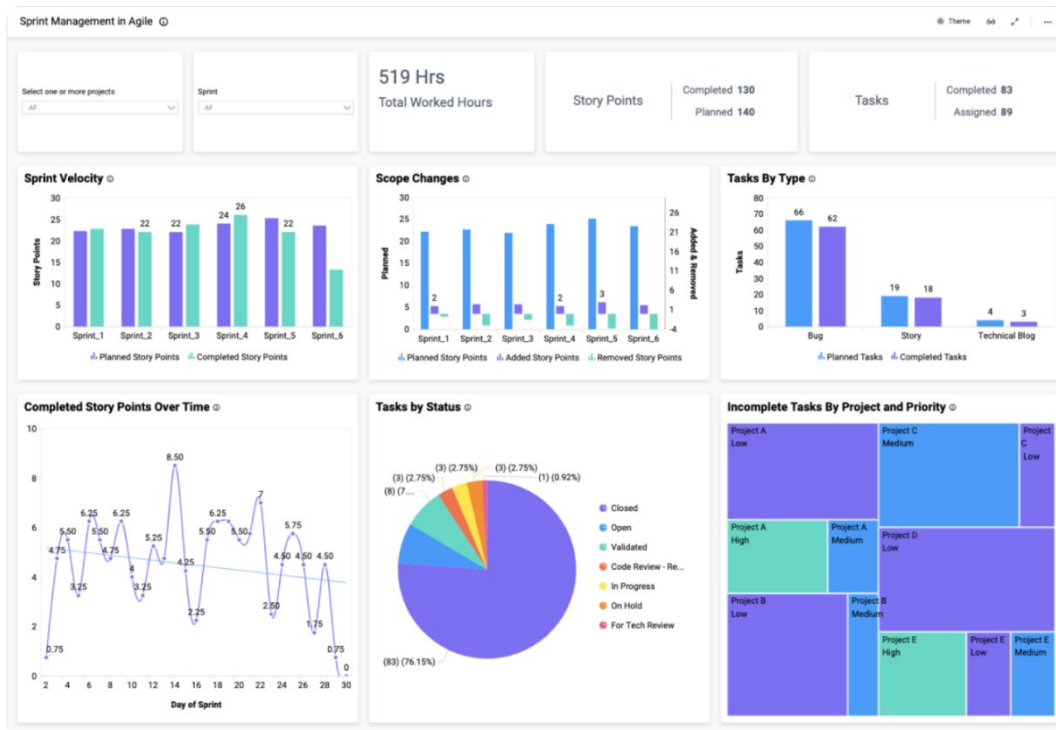
...and many others

Product Strategy Canvas

The canvas should be updated everytime we gain better understanding about our business, our product and our customers.

Product Vision					
Business Strategy What business problem(s) have we identified that needs to be solved? What is our strategy to solve those problems? <i>Note: Work together with the business and senior management to craft the business strategy statement.</i>		Objective The objective we want to achieve before the next OKR Review. Key Results The key measurable indicators as a proof that we have met our objective. <i>Hint: look at our value driven metrics box.</i>		Customers Who are the people (or group) that will pay us or give funding to us to solve this problem?	
Value Driven Metrics What value driven metrics, as an indicator of success, for our strategy should we track continuously? <i>NOT on-scope, on-time and on-budget.</i> <i>Note: check out www.scrum.org/bsm as a source of inspiration.</i>		Consumers Who are the people we've identified facing this problem(s) and will potentially use the solution we will deliver?		Customer Value What value will the customer get if this problem is solved?	
North Star Metric There should only one North Star Metric. It is the only metric that captures one of all of the other value driven metrics.		SMART Sprint Goal <i>Specific, Measurable, Attainable, Realistic, Time-bound</i> Note: craft the Sprint Goal with the development team. Sprint # Sprint Review Date		SMART Sprint Goal <i>Specific, Measurable, Attainable, Realistic, Time-bound</i> Note: craft the Sprint Goal with the development team. Sprint # Sprint Review Date	
Possible Solution Options What are some possible solutions (features, products, enhancements) that we know as of today? These options are part of the Product Backlog. <i>Note: collaborate with the development team to create the known possible solution options.</i>		Hypotheses What are the things we have listed in previous boxes may not be true, has no empirical data and needs to be validated? Quantify the risk for each hypotheses. <i>These hypothesis are part of the Product Backlog.</i> <i>Note: collaborate with the development team to create the hypothesis.</i>		Consumer Outcome What behavioural changes that we expect from the consumer after we solve the problem?	

Product Strategy Canvas (V 0.9.3) by Joshua Partogi. Inspired by Lean UX Canvas by Jeff Gothelf. Download this canvas from: <http://agilitypath.com.au/tools>.



Velocity Calculations

Fixed date – Today = # of days

of days / 7 = # of weeks

of weeks / Sprint duration = **# of Sprints**

Velocity x # of Sprints =
Estimated Scope

Example:

Developers that can do 40 items / Sprint on average must release by November 30. That date is 6 Sprints from today. How much scope (approximately) can the Developers complete?

Fixed Scope ÷ Velocity =
Estimated # of Sprints

Example:

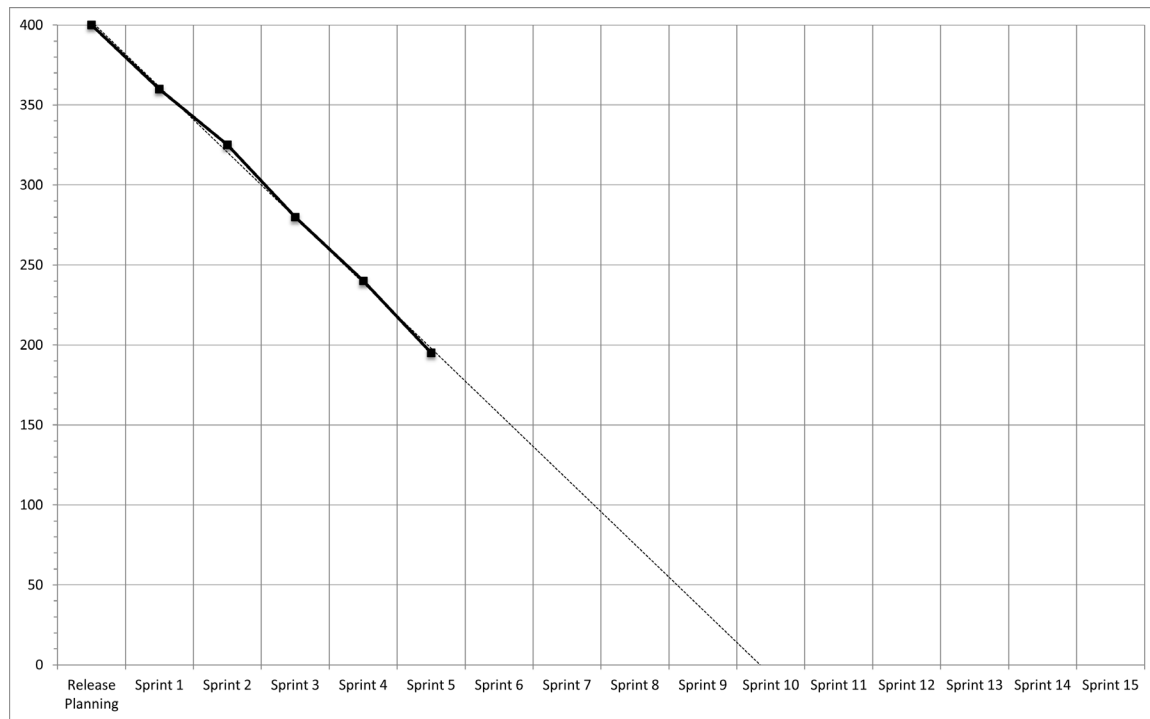
A Product Owner indicates that the Release will happen when the MVP is ready, which is represented by 240 Story Points worth of Product Backlog Items. The Developers' velocity is 60 Story Points / Sprint. What is the estimated delivery date if today is February 1st our Sprints are 3 weeks in duration?

of Sprints x Sprint duration = # of weeks

of weeks x 7 = # of days

Today + # of days = **delivery date (estimated)**

Release Burndown



In this example, the team starts with 400 Story Points of scope identified (Product Backlog Items). The Developers begin working on these items and based on an average Velocity of 41 Story Points / Sprint, they will be done in 10 Sprints.

**Release Burndowns should
be updated after every Sprint.**

CSPO-1.3 List at least three techniques to gather information or insights from stakeholders

- **Affinity grouping**
- **Monopoly voting**
- **Prune the Product Tree**
Innovation Games
- **Ask Scrum Master to help with Facilitation**

**CSPO-1.5 Identify at least two ways to overcome the challenges
of being a Product Owner for multiple Scrum Teams**

- **Ensure that the product / work is
closely aligned**
- **Be transparent with leadership if
serving multiple teams is simply
NOT realistic**
- **Enlist the help of business analysts,
team leads, and the Scrum Master to
help with refinement and other PO
activities**

Goal Setting and Planning



Product planning, as with all other things in life, is like traveling through a forest in the fog or driving at night in a car with the headlights on; the further out we try to see, the less clarity and certainty we have...

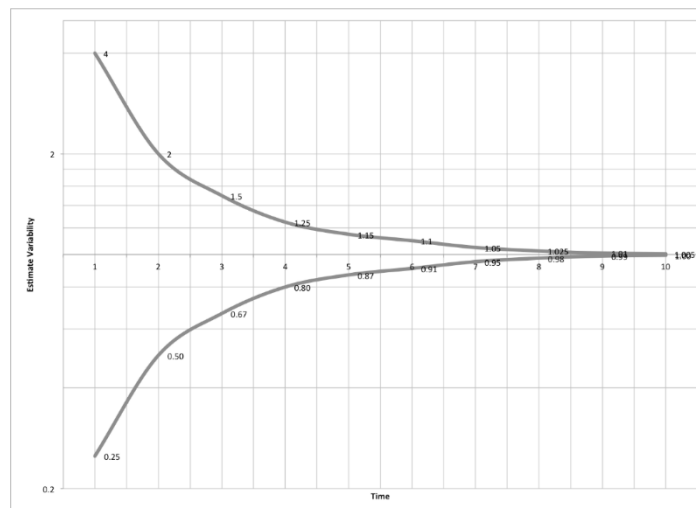
In Scrum, we acknowledge this reality rather than trying to pretend we can know for certain what the future holds.

**"The more precise the measurement of position,
the more imprecise the measurement of momentum,
and vice versa."**

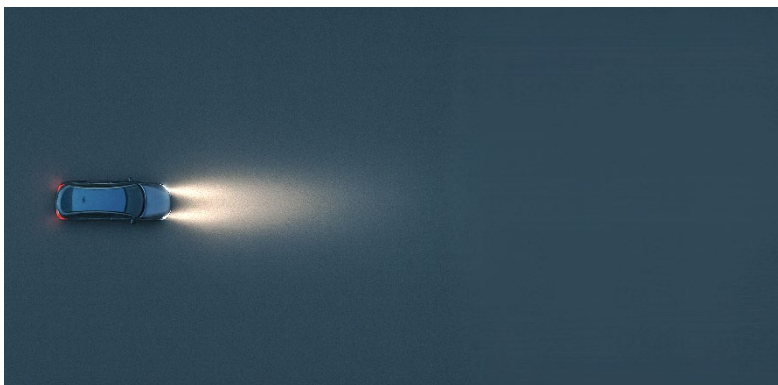
*-Werner Heisenberg
The Uncertainty Principle*

Planning (aka estimation or forecasting) is ALWAYS
inaccurate and imprecise.
We are trying to **GUESS** what will happen in the future.

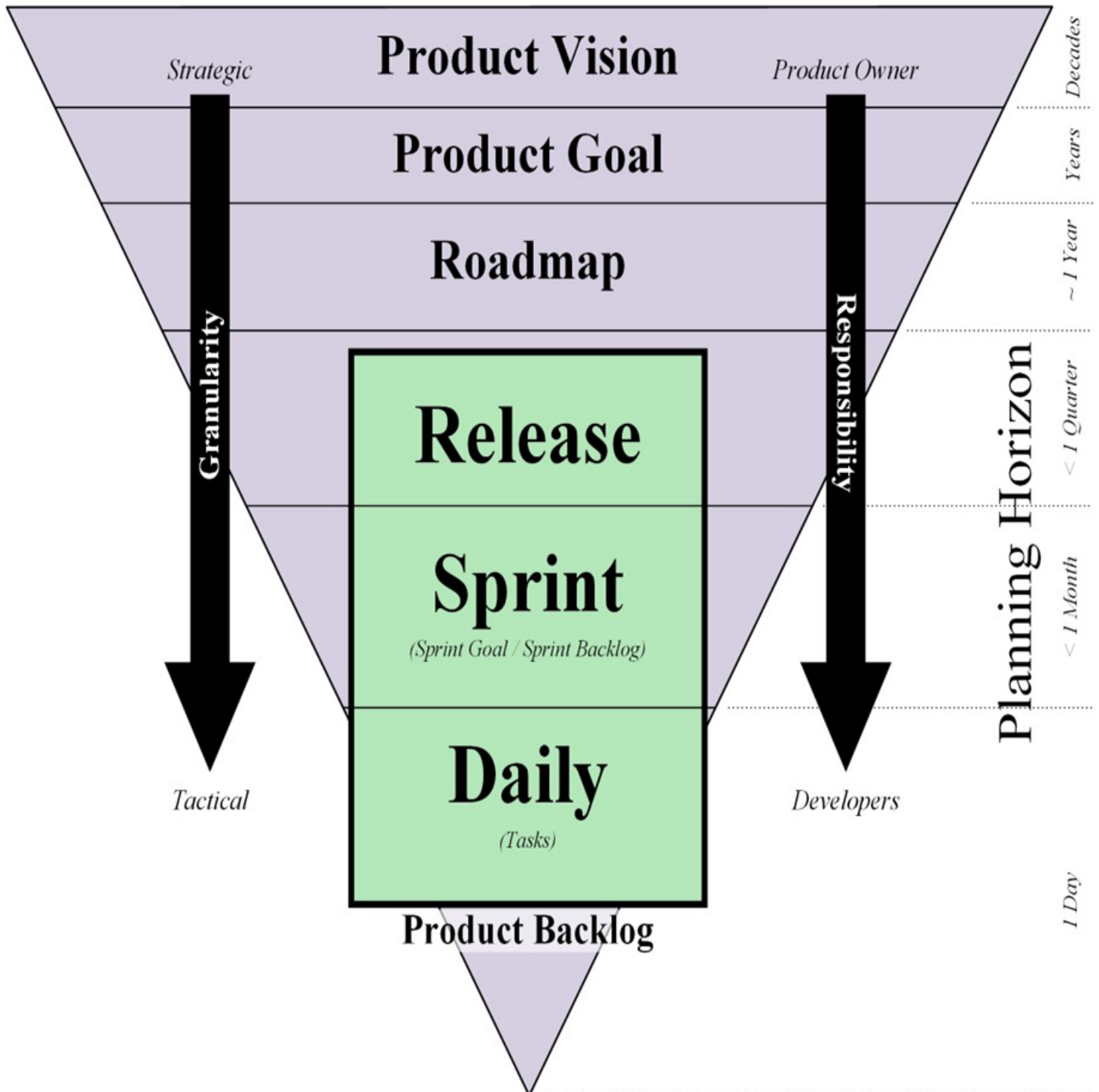
Cone of Uncertainty



The further out the planning horizon, the more variance there is in our plans TODAY; i.e. when we estimate things that are further out, our estimates become less and less accurate.



Six-level Model of Agile Planning



CSPO-2.1 Explain the relationship between a product vision and a Product Goal

Communicate the purpose of a product idea by describing overall the problem being solved, who is most affected by the problem, how the Team's efforts will improve the situation, and how that solution's effectiveness will be evaluated

Product Vision

(aka "Elevator Pitch Template")

"FOR *(target customer)*
WHO *(statement of the need)*
THE *(product name)* **is a** *(product category)*
THAT *(product key benefit, compelling reason to buy).*
UNLIKE *(primary competitive alternative),*
OUR PRODUCT *(final statement of primary differentiation)"*

Source:

Geoffrey Moore - "Crossing the Chasm"

Sample Visions

Ikea

To create a better every day life for the many people.

Microsoft

Empower every person and every organization on the planet to achieve more.

Nike

Bring inspiration and innovation to every athlete* in the world. (*If you have a body, you are an athlete.)

Ford

People working together as a lean, global enterprise to make people's lives better through automotive and mobility leadership.

Avon

To be the company that best understands and satisfies the product, service and self-fulfillment needs of women—globally.

Honda – in 1970

We will destroy Yamaha.

Nike – in 1960s

Crush Adidas.

Philip Morris – in 1950s

Knock off RJR as the number one tobacco company in the world.

Stanford University – in the past

To become the Harvard of the West.

Reach for Success – in the past

To become the next Tony Robbins in self development.

Apple

To produce high-quality, low cost, easy to use products that incorporate high technology for the individual.

Giro Sport Design

To make sure that riding is the best part of a great life.

Tesla

To accelerate the world's transition to sustainable energy.

Sony

To be a company that inspires and fulfills your curiosity.

Facebook

To give people the power to share and make the world more open and connected.

Walmart

To give customers a wide assortment of their favorite products, Every Day Low Prices, guaranteed satisfaction, friendly service, convenient hours (24 hours, 7 days a week) and a great online shopping experience.

Life is Good

To spread the power of optimism.

sweetgreen

To inspire healthier communities by connecting people to real food.

Patagonia

Build the best product, cause no unnecessary harm, use business to inspire and implement solutions to the environmental crisis.

American Express

We work hard every day to make American Express the world's most respected service brand.

InvisionApp

Question Assumptions

Honest Tea

To create and promote great-tasting, healthy, organic beverages.

Nordstrom

To give customers the most compelling shopping experience possible.

JetBlue

To inspire humanity – both in the air and on the ground.

Workday

To put people at the center of enterprise software.

Prezi

To reinvent how people share knowledge, tell stories, and inspire their audiences to act.

Tesla

To accelerate the world's transition to sustainable energy.

Invisible Children

To end violence and exploitation facing our world's most isolated and vulnerable communities.

TED

Spread ideas.

Alzheimer's Association

A world without Alzheimer's disease.

Teach for America

One day, all children in this nation will have the opportunity to attain an excellent education.

Creative Commons

Realizing the full potential of the internet — universal access to research and education, full participation in culture — to drive a new era of development, growth, and productivity.

Microsoft (at its founding)

A computer on every desk and in every home.

LinkedIn

Create economic opportunity for every member of the global workforce.

Facebook

Connect with friends and the world around you on Facebook.

Southwest

To be the world's most loved, most efficient, and most profitable airline.

Dunkin'

To be always the desired place for great coffee beverages and delicious complementary doughnuts & bakery products to enjoy with family and friends.

CSPO-2.2 Practice the creation of a Product Goal

Product Goal





	What is your purpose for creating the product? Which positive change should it bring about?		
 TARGET GROUP Which market or market segment does the product address? Who are the target customers and users?	 NEEDS What problem does the product solve? Which benefit does it provide?	 PRODUCT What product is it? What makes it stand out? Is it feasible to develop the product?	 BUSINESS GOALS How is the product going to benefit the company? What are the business goals?

www.romanpichler.com

Template version 05/17

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License

SMART Goals

S – Specific

M – Measurable

A – Achievable

R – Relevant

T – Time-boxed

A Product Goal can roughly be equated to a Release Goal since it represents an increment of value that we are delivering to a customer.

CSPO-2.4 List the components of a product plan or forecast with stakeholders

- **Roadmaps are commonly used as a traditional product plan**
- **The roadmap is intended to be dynamic and amenable so that customer feedback can shape the product over time.**
- **The roadmap is intended to include high-level milestones, themes, and ideas of WHAT will deliver value to the customer.**
- **The roadmap focuses on the problem domain.**
- **The roadmap is NOT intended to be a comprehensive list of features.**
- **The roadmap is NOT intended to delve into the HOW or solution domain.**

GO (goal oriented) Roadmap

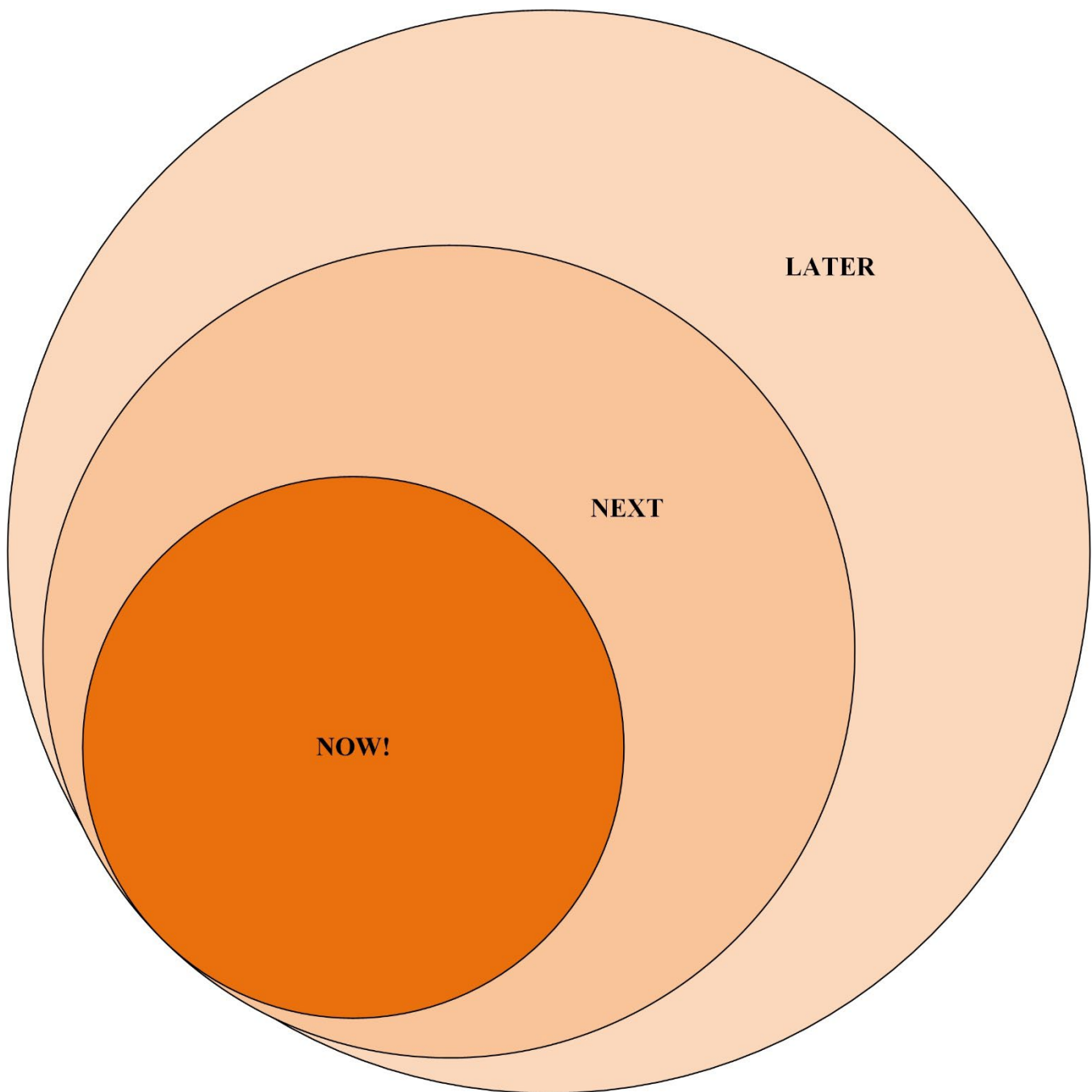
THE GO PRODUCT ROADMAP



DATE The date or time frame when a goal should be met.					
NAME The name of the new release.					
GOAL The outcome or benefit you want to achieve.					
FEATURES The high-level features required to meet the goal.					
METRICS The measures to determine if the goal has been met.					



The Now, Next, Later Product Roadmap



CSPO-2.5 Describe how to plan a product release

Product Backlog exists and contains enough value to justify a Release.

A Release Goal is created.

Product Owner determines the state of constraints in the “iron triangle”:

One can be FIXED (committed), one may be FIRM (targeted) and the last one FLEXIBLE (unlimited).

The more flexible the third constraint, the more probable the second constraint will be.

Product Owner works together with the Developers and Scrum Master to ensure PBIs are understood and include Acceptance Criteria and Estimates.

Velocity is used to validate the feasibility of the Release Goal.

Developers provide information and concerns about dependencies and ask any clarifying questions

Scrum Team discusses if the Release makes sense and is reasonable

Minimum Viable Product (MVP)

Value: learning

“That version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort.”

-Eric Ries, The Lean Startup

Minimum Marketable Features (MMF)

Value: ultimate delivery to customers

A small, self-contained feature that can be developed quickly and that delivers significant value to the user.

(Source: Agile Alliance)

CSPO-2.3 Creating a Sprint Goal with a Scrum Team

ROMAN'S SPRINT GOAL TEMPLATE



 PRODUCT Name the product and the product goal.	 SPRINT State the sprint number or name.
 GOAL Describe why it is worthwhile to run the sprint. What outcome should be achieved? For instance, address a risk, test an assumption, or deliver a feature. Capture the reason for running the sprint, not the output.	
 METHOD Explain how the goal will be met. Which artefact, validation technique, and test group will be used? For example, paper prototype, spike, shippable product increment; product demo, usability test, A/B test; users, customers and/or internal stakeholders.	
 METRICS State how you will determine if the goal has been met. For instance, at least three out of five users can successfully carry out the usability test in less than two minutes.	

www.romanpichler.com
Version 03/2023

This template is licensed under a Creative Commons
Attribution-ShareAlike 4.0 Unported license.



Understanding Customers and Users

CSPO-3.1 Describe how a Product Owner includes product discovery
and validation in their work

- **The low usage rate of delivered features in waterfall**
Gartner / Standish Reports
- **The high failure rate of start-ups**
Crossing the Chasm
- **The impact of cognitive bias**
AYSO model of decision-making

Look in the mirror...

**THE BUCK STOPS
HERE !!**

CSPO-3.2 Illustrate at least one approach for segmenting customers and users

CSPO-3.3 Practice at least one technique to address conflicting customer (or user) needs

Customer Types:

- **Geographic**
- **Psychographic**
- **Socio-Economic**
- **Role based**

...and many others

Why Personas?

Our Customer:

- **Male**
- **Born in 1948**
- **Raised in the UK**
- **Married twice**
- **Lives in a castle**
- **Wealthy and famous**



Charles

- Male
- Born in 1948
- Raised in the UK
- Married twice
- Lives in a castle
- Wealthy & famous



Ozzy

- Male
- Born in 1948
- Raised in the UK
- Married twice
- Lives in a castle
- Wealthy & famous

Do **King Charles** and **Ozzy Osbourne**
have different interests?
Needs? Motivations? Desires?

NOTE: Ozzy Osbourne passed away on July 22, 2025. He made a significant impact on the music industry throughout his lifetime. I have been using this example for many years and made the decision to continue using it with respect to Ozzy's memory.

Creating Personas

BDT

Persona type _____ Date _____

Author _____

PERSONA CANVAS

☐ PRIMARY ☐ SECONDARY

“ Statement/behaviour ”

Name _____
 Age _____
 Occupation _____
 Location _____
 Status _____

Bio _____

Personality EXTROVERT ☐ INTROVERT ☐ SENSING ☐ INTUITION ☐ THINKING ☐ FEELING ☐ JUDGING ☐ PERCEIVING ☐ PASSIVE ☐ ACTIVE ☐	Technology expertise IT & INTERNET ☐ SOFTWARES ☐ MOBILE APPS ☐ SOCIAL NETWORKS ☐ INTERNET OF THINGS & VIRTUAL REALITY ☐	Where to reach me TRADITIONAL ADS & DIRECT MAIL ☐ ONLINE ADS & EMAIL ☐ BLOGS & SOCIAL MEDIA ☐ REFERRALS ☐ PHYSICAL LOCATION/EVENTS ☐	Goals
What makes me get involved INCENTIVE ☐ FEAR ☐ ACHIEVEMENT ☐ GROWTH ☐ POWER ☐ SOCIAL ☐	Pain points 	Relationship with your product/service ← FREQUENCY OF USE → VALUE	
Fave brands/apps/influencers 	Top reason to use your product/service (USP) 	Dealbreaker 	
Devices and platforms 			

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License. To view a copy of this license visit: <https://creativecommons.org/licenses/by-sa/3.0/>



www.businessdesigntools.com

romanpichler

ROMAN'S PERSONA TEMPLATE

PICTURE & NAME	DETAILS	GOAL
What does the persona look like? What is its name? Choose a realistic and believable picture and name.	What are the persona's relevant characteristics and behaviours? For instance, demographics, such as age, gender, occupation, and income; psychographics, including lifestyle, social class, and personality; and behavioural attributes like usage patterns, attitudes, and brand loyalty. Only list relevant details.	What problem does the persona want to solve or which benefit does the character seek? Why would the persona want to use or buy the product?

www.romanpichler.com

Template version 04/17

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License

CC BY SA



Nerdy Nina

"The book is way better than the movie!"

#booklover
#bookaddict
#booknerdproblems

DEMOGRAPHICS

Age: 25
Location: Sao Paulo, Brazil
Education: Software Engineer
Job: Q/A at Indie Game Company
Family: Lives with her boyfriend

TECH

Internet
Social Networks
Messaging
Games
Online Shopping



GOALS

- Discovering new books / authors to read
- Finding unique stories
- Cataloging book collection

FRUSTRATIONS

- Keeping track of different series
- Forgetting a book launch date
- Finding space for more books

READING HABITS

- Fast pace reader
- Never lends books
- Likes hardcovers and boxed collections
- Pre-order books to get them first
- Reads eBooks, but prefer physical copies
- Always finishes a book
- Loves binge reading and re-reading

FAVORITE BOOKS



American Gods
Neil Gaiman



Harry Potter
J.K. Rowling



Ready Player
One

Rory

Web developer

ABOUT

Age 32
Salary 29K€
Location Barcelona
Status with couple

GOALS

Always get the best deals
Escape from cities
Telework

FRUSTRATIONS

Have no plans
To lose money
No free wifi

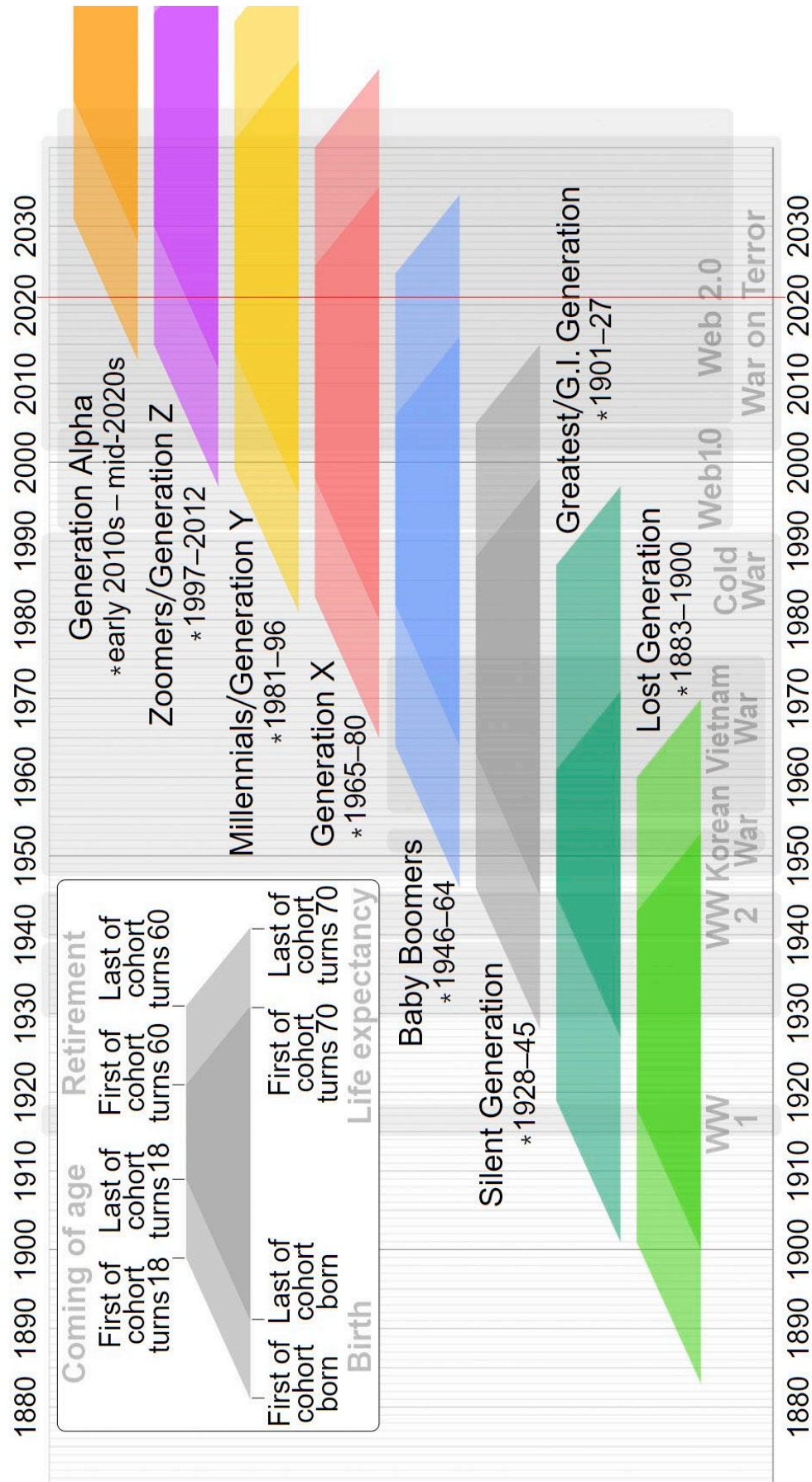
PERSONALITY

Extrovert ———— Introvert
Sensing ———— Intuition
Thinking ———— Feeling
Judging ———— Perceiving

TECHNOLOGY

IT and Internet ●●●●●●●●●●
Software ●●●●●●●●●●
Mobile Apps ●●●●●●●●●●
Social Networks ●●●●●●●●●●

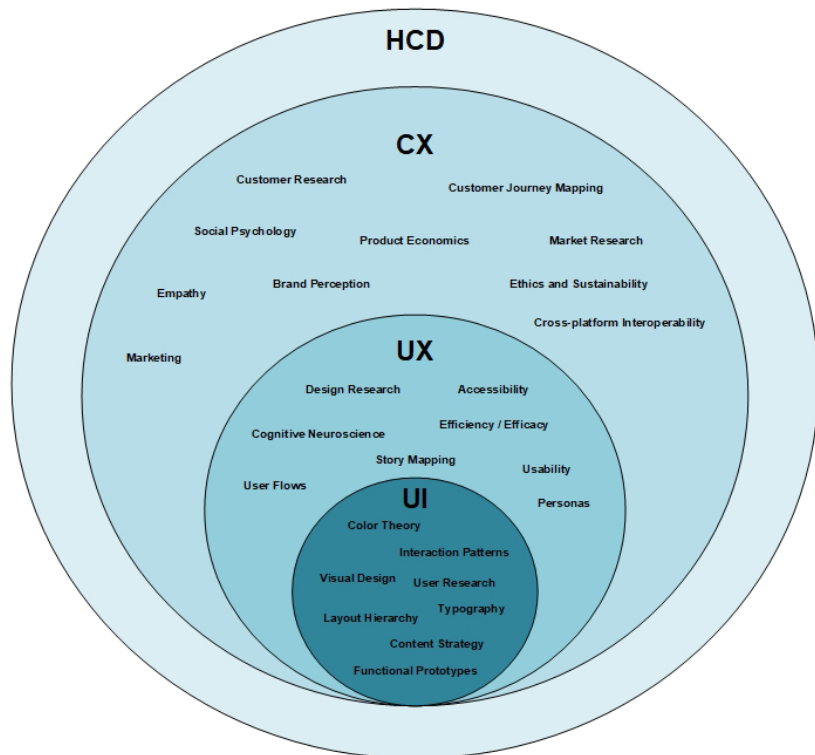




CSPO-3.4 Identify how at least three aspects of product discovery contribute to successful product outcomes

Human Centered Design (HCD) *aka User Centered Design (UCD)*

- HCD is a subset of **Design thinking (DT)**, which is focused on the questions “What is the right problem to solve? What are we trying to accomplish?”
- **Lean thinking (LT)** focuses on the question “What would be the most valuable solution to the problem?”
- **Agile thinking (AT)** focuses on the question “What is the most effective way to build and deliver the solution?”
- All three ways of thinking are being employed throughout the product lifecycle, NOT just in discrete phases.
- Hypotheses (assumptions) initiated with HCD are framed into experiments in LT and the experiments are conducted in AT where we learn the results and outcome of the hypotheses.
- Theory guides practical applications. Practical applications inform theory.
- The interaction of these feedback loops across an org structure is part of systems thinking (managerial cybernetics).



Lean Thinking

Agile Thinking

CSPO-3.5 List at least three approaches to connect Developers directly to customers and users

Customer Observation

Direct:

- Job Shadowing
- Collaborative Customer Games
E.g. Innovation Games, Lego Serious Play
- Interviews
- Focus Groups

Indirect:

- Surveys
- Website trackers, cookies, scripts, etc.
- Sending usage metrics back to the company
- Video cameras recording behaviors (*iaw all privacy laws, of course*)

IO Psychology

Working with the Product Backlog

Product Backlog Refinement

- Adding items
- Removing items
- Defining Acceptance Criteria
- Splitting items into smaller pieces
- Re-ordering items to indicate priority
 - Mitigating dependencies

Any changes to the Product Backlog are

REFINEMENT

We REFINE the Product Backlog.

We do not GROOM it.

"Grooming" is gross.

groom

- To clean and brush (an animal).
- To remove dirt and parasites from the skin, fur, or feathers of (another animal).
- To win the confidence of (a victim) in order to commit sexual assault on him or her

refine

- To become free of impurities.
- To acquire polish or elegance.
- To use precise distinctions and subtlety in thought or speech.
- To make or become free from coarse characteristics; make or become elegant or polished
- To make more fine, subtle, or precise.
- To become pure.
- To become more elegant or polished.
- To make fine distinctions in thought or language.



CSPO-5.1 Describe the relationship between outcome and output

- **Output is a measure of what was built**

- **Outcome is how that output impacts users and customers**

We could spend 10 hours producing something the customer is willing to pay \$1000 for and our ROI would be \$100/hour.

However, what if we could only spend 1 hour producing something that the customer would be equally happy with? Our ROI would be \$1000/hour.

Minimizing OUTPUT and maximizing OUTCOMES
boosts ROI (PROFIT) just as maximizing VALUE
and minimizing WASTE also increases PROFIT.

Asking ourselves "What is absolutely necessary?" in terms of documentation, process, meetings, etc. helps us to understand what we can get rid of in terms of wasteful overhead that doesn't add value to the customer / users.

CSPO-5.2 List at least three actions the Product Owner could take to maximize outcomes and impact while minimizing outputs

- **Who wants it?**
- **Why do they need it?**



*"It's what you **NEED**."*
Twilight Zone, "What You Need"

**Quite often, what the customer WANTS
is not what they NEED.**

CSPO-5.5 List at least three techniques to measure value

Product

NPS

(Net Promoter Score)

Total % of promoters – total % of detractors = net promoter score

<https://www.qualtrics.com/experience-management/customer/measure-nps/>

Happiness Metric



Trust Pilot

Net Present Value

$$NPV = \sum_{t=0}^n \frac{R_t}{(1+i)^t}$$

<https://www.investopedia.com/terms/n/npv.asp>

Product Backlog Items

RICE

(Reach x Impact x Confidence) / Effort

<https://www.productplan.com/glossary/rice-scoring-model/>

WSJF

(Weighted Shortest Job First)

<https://www.scaledagileframework.com/wsjf/>

Kano Analysis

<https://www.qualtrics.com/experience-management/research/kano-analysis/>

CSPO-5.4 Describe value from the perspective of at least three different stakeholder groups

Customer / Users

Business stakeholders

Developers

Investors

Sales / Marketing

CSPO-5.3 Define at least three terms related to product economics

Iron Triangle / Triple Constraint

Scope • Cost • Time / Date



"Jim Jarmusch once told me 'Fast, Cheap, and Good...pick two. If it's fast and cheap it won't be good. If it's cheap and good it won't be fast. If it's fast and good it won't be cheap.'"

-Tom Waits

Source: "Tom Waits Spills the Beans to Tom Waits," Sound Effects Blog, Boston Globe, May 22, 2008.

Product Backlog Item Types

A PBI (product backlog item) is first and foremost an increment of value that we intend to deliver to a customer.

Examples

- **Features**

(increments of value)

- **Enhancement Requests**

- **Production Defect Fixes**

- **Technical Debt**

(Optional, Unavoidable, Avoidable)

Sometimes people include **Compliance Requirements** as PBIs. MOST compliance concerns should be handled by the DoD criteria. Occasionally, there might be something that is broad, general, and applicable to the entire product vs. pieces that can be updated at the PBI level.

Product Backlog Item Formats/Templates

Examples

Connextra
(aka “user stories”)

Job Stories

Agile Lifecycle Management (ALM)
Tool Templates
(e.g. JIRA, ADO, Rally, VersionOne, etc.)

Connextra *(aka User Stories)*

One popular template for Product Backlog Items

"As a *(who)* ,

I would like *(what)* ,

So that, *(why)* ."

This format is good for differentiating behaviors among different user types or groups. There is very limited value in using this format for behaviors which apply to ALL users.

“A user story is a promise of a conversation.”

-Alistair Cockburn

Job Stories

Yet another kind of Product Backlog Item

"When (situation or condition) ,

I want (what) ,

So that (expected outcome / why) ."

The idea behind Job Stories (or rather the argument raised by the proponents) is that there are too many assumptions around personas and desired actions. "How do we know this is the best action to take? Maybe there's a better way..." they say about the first two clauses of User Story format.

However, I don't particularly see that their proposed format really solves this. In fact, both formats fall short in that the actor is missing in Job Stories and the situation is absent from User Stories. So, perhaps a hybrid?

But then, Use Cases, right???

Sources:

<https://blog.intercom.io/using-job-stories-design-features-ui-ux/>

<https://medium.com/the-job-to-be-done/replacing-the-user-story-with-the-job-story-af7cdee10c27#.4f0g4jwiq>

ALM Tool Templates

EXAMPLE: Default JIRA Fields Available

Field	Description
Project	The parent project to which the issue belongs.
Key	A unique identifier for this issue, in the example above: ANGRY-304. (The characters to the left of the hyphen represent the project to which this issue belongs.)
Summary	A brief one-line summary of the issue. For example, "Red Angry Nerd is scary."
Type	See below for a list of types.
Status	The stage the issue is currently at in its lifecycle (workflow). See below for a list of statuses.
Priority	The importance of the issue in relation to other issues. (See below for a list of priorities).
Resolution	A record of the issue's resolution, if the issue has been resolved or closed. (See below for a list of resolutions).
Affects Version(s) <i>(if applicable)</i>	Project version(s) for which the issue is (or was) manifesting.
Fix Version(s) <i>(if applicable)</i>	Project version(s) in which the issue was (or will be) fixed.
Component(s) <i>(if applicable)</i>	Project component(s) to which this issue relates.
Labels <i>(if applicable)</i>	Labels to which this issue relates.
Environment <i>(if applicable)</i>	The hardware or software environment to which the issue relates.
Description	A detailed description of the issue.
Links	A list of links to related issues. (Strikethrough text, ~~like this~~ , indicates that an issue has been resolved.)
Assignee	The person to whom the issue is currently assigned. Note that you cannot assign issues to a user group.
Reporter	The person who entered the issue into the system.
Votes	The number shown indicates how many votes this issue has.
Watchers	Number shown indicates how many people are watching this issue.
Due <i>(if applicable)</i>	The date by which this issue is scheduled to be completed.
Created	The time and date on which this issue was entered into Jira.
Updated	The time and date on which this issue was last edited.
Resolved	The time and date on which this issue was resolved.
Estimate	The Original Estimate of the total amount of time required to resolve the issue, as estimated when the issue was created.
Remaining	The Remaining Estimate, i.e. the current estimate of the remaining amount of time required to resolve the issue.
Logged	The sum of the Time Spent from each of the individual work logs for this issue.
Development	If you use Bitbucket to manage your code repositories, you can create code branches in your code development tools directly from Jira issues. See Integrating with development tools for details.
Agile	Lets you view your issue on your Scrum or Kanban board.
Service Desk	Lets you view request participants and view the equivalent request in the customer portal

Acceptance Criteria

ALL PBIs need to have at least ONE Acceptance Criteria
and usually have MANY.

Behavior Driven Development (BDD)

Given... When... Then...

(Gherkin syntax for Cucumber)

Example:

Feature: Account Holder withdraws cash

Scenario: Account has sufficient funds

Given the account balance is \$100

And the card is valid

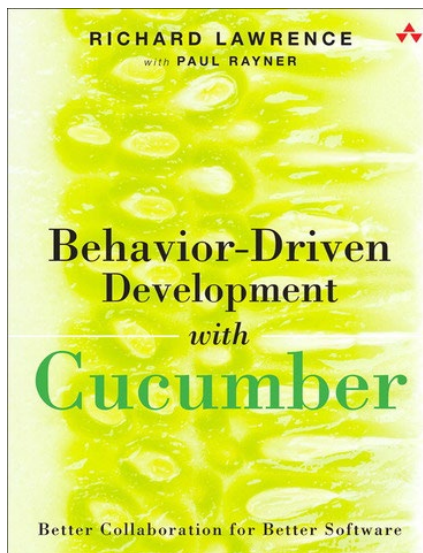
And the machine contains enough money

When the Account Holder requests \$20

Then the ATM should dispense \$20

And the account balance should be \$80

And the card should be returned



Without Acceptance Criteria, there will not be a clear understanding of whether the item is DONE or not.

Definition of Done

The Definition of Done (DOD) is decided, agreed upon, and updated as needed by each Scrum Team. The DOD includes all tasks and activities necessary to ensure that each Product Backlog Item results in a Potentially Shippable Product Increment.

Two key components of the **Definition of Done** are:

Definition of Done	Acceptance Criteria	represents building the RIGHT PRODUCT	as defined by the Product Owner
	Examples: - Functional Requirements “happy path” - Business Rules - Ability to do something - Saves customer/user time - Saves customer/user money - Solves a problem - Provides information to make better decisions - etc.		
	Engineering Criteria	represents building the PRODUCT RIGHT	as defined by the Developers
	Examples: - Unit Testing - Refactoring - Peer Reviews - Automation (Build, Deployment, Testing) - Documentation - Functional Testing - Performance Testing - Deployment Scripts - Normalize DB Tables - Follow Branding Standards - Regulatory Compliance / Governance (and MANY other technical / non-functional things)		

Q: What format should we use for OUR Product Backlog Items?

A: Whatever the Scrum Team agrees on in terms of capturing enough detail to GET TO WORK creating Customer Value ASAP!!

INVEST in Good Stories, and SMART Tasks

by Bill Wake

August 17, 2003

In XP, we think of requirements of coming in the form of user stories. It would be easy to mistake the story card for the "whole story," but Ron Jeffries points out that stories in XP have three components: Cards (their physical medium), Conversation (the discussion surrounding them), and Confirmation (tests that verify them).

A *pidgin language* is a simplified language, usually used for trade, that allows people who can't communicate in their native language to nonetheless work together. User stories act like this. We don't expect customers or users to view the system the same way that programmers do; stories act as a pidgin language where both sides can agree enough to work together effectively.

But what are characteristics of a good story? The acronym "INVEST" can remind you that good stories are:

I – Independent
N – Negotiable
V – Valuable
E – Estimable
S – Small
T – Testable

Independent

Stories are easiest to work with if they are *independent*. That is, we'd like them to not overlap in concept, and we'd like to be able to schedule and implement them in any order.

We can't always achieve this; once in a while we may say things like "3 points for the first report, then 1 point for each of the others."

Negotiable... and Negotiated

A good story is *negotiable*. It is not an explicit contract for features; rather, details will be co-created by the customer and programmer during development. A good story captures the essence, not the details.

Over time, the card may acquire notes, test ideas, and so on, but we don't need these to prioritize or schedule stories.

Valuable

A story needs to be *valuable*. We don't care about value to just anybody; it needs to be valuable to the customer. Developers may have (legitimate) concerns, but these framed in a way that makes the customer perceive them as important.

This is especially an issue when splitting stories. Think of a whole story as a multi-layer cake, e.g., a network layer, a persistence layer, a logic layer, and a presentation layer. When we split a story, we're serving up only part of that cake. We want to give the customer the essence of the whole cake, and the best way is to slice vertically through the layers. Developers often have an inclination to work on only one layer at a time (and get it "right"); but a full database layer (for example) has little value to the customer if there's no presentation layer.

Making each slice valuable to the customer supports XP's pay-as-you-go attitude toward infrastructure.

Estimable

A good story can be *estimated*. We don't need an exact estimate, but just enough to help the customer rank and schedule the story's implementation. Being estimable is partly a function of being negotiated, as it's hard to estimate a story we don't understand. It is also a function of size: bigger stories are harder to estimate. Finally, it's a function of the team: what's easy to estimate will vary depending on the team's experience. (Sometimes a team may have to split a story into a (time-boxed) "spike" that will give the team enough information to make a decent estimate, and the rest of the story that will actually implement the desired feature.)

Small

Good stories tend to be *small*. Stories typically represent at most a few person-weeks worth of work. (Some teams restrict them to a few person-days of work.) Above this size, and it seems to be too hard to know what's in the story's scope. Saying, "it would take me more than a month" often implicitly adds, "as I don't understand what-all it would entail." Smaller stories tend to get more accurate estimates. Story descriptions can be small too (and putting them on an index card helps make that happen). Alistair Cockburn described the cards as tokens promising a future conversation. Remember, the details can be elaborated through conversations with the customer.

Testable

A good story is *testable*. Writing a story card carries an implicit promise: "I understand what I want well enough that I *could* write a test for it." Several teams have reported that by requiring customer tests before implementing a story, the team is more productive. "Testability" has always been a characteristic of good requirements; actually writing the tests early helps us know whether this goal is met.

If a customer doesn't know how to test something, this may indicate that the story isn't clear enough, or that it doesn't reflect something valuable to them, or that the customer just needs help in testing.

A team can treat non-functional requirements (such as performance and usability) as things that need to be tested. Figure out how to operationalize these tests will help the team learn the true needs.

For all these attributes, the feedback cycle of proposing, estimating, and implementing stories will help teach the team what it needs to know.

SMART Tasks

There is an acronym for creating effective goals: "SMART" –

S – Specific

M – Measurable

A – Achievable

R – Relevant

T – Time-boxed

(There are a lot of variations in what the letters stand for.) These are good characteristics for tasks as well.

Specific

A task needs to be *specific* enough that everyone can understand what's involved in it. This helps keep other tasks from overlapping, and helps people understand whether the tasks add up to the full story.

Measurable

The key *measure* is, "can we mark it as done?" The team needs to agree on what that means, but it should include "does what it is intended to," "tests are included," and "the code has been refactored."

Achievable

The task owner should expect to be able to *achieve* a task. XP teams have a rule that anybody can ask for help whenever they need it; this certainly includes ensuring that task owners are up to the job.

Relevant

Every task should be *relevant*, contributing to the story at hand. Stories are broken into tasks for the benefit of developers, but a customer should still be able to expect that every task can be explained and justified.

Time-Boxed

A task should be *time-boxed*: limited to a specific duration. This doesn't need to be a formal estimate in hours or days, but there should be an expectation so people know when they should seek help. If a task is harder than expected, the team needs to know it must split the task, change players, or do something to help the task (and story) get done.

Conclusion

As you discuss stories, write cards, and split stories, the INVEST acronym can help remind you of characteristics of good stories. When creating a task plan, applying the SMART acronym can improve your tasks.

[I developed the INVEST acronym, and wrote this article in April and August, 2003. Thanks to Mike Cohn for his encouragement and feedback.]

Postscript – Added 2-16-2011

I've been asked a few times where the INVEST acronym came from. I consciously developed it: I sat down and wrote every attribute I could think of applying to good stories: independent, small, right-sized, communicative, important, isolated, etc. This gave me a page full of words. Unfortunately, I haven't kept that list.

Then I grouped the words into clusters:

Isolated, independent, separate, distinct, non-overlapping, ...

Important, valuable, useful, ...

Discrete, triggered, explicit, ...

...

The categories were a little fuzzy; I had about ten.

I identified "centers"; words that captured the essence of their category. Some clusters had two or three plausible centers. That was ok, as I just wanted their first letter for the acronym.

Then it was scramble time: take the three or four clusters that had to be there, plus some of the less important ones, and scramble the initials to find a word that fit. I wanted a word that had 4-6 letters, with no repeats.

I tried a lot of combinations. I remember at one point that if I had a G I could make VESTIGE or VESTING. Having VESTIN_, I realized I could turn it around to get INVEST, which sounded much better than VESTIGE:) So INVEST won, and it's been popular enough that I'm sure I made the right choice.



Story Splitting



Story Splitting Cheat Sheet

The INVEST Model

Stories should be: Independent, Negotiable, Valuable, Estimable, Small, and Testable.

Patterns for Splitting Stories

Workflow Steps

As a content manager, I can publish a news story to the corporate website.

...I can publish a news story directly to the corporate website.
 ...I can publish a news story with editor review.
 ...I can publish a news story with legal review.

Business Rule Variations

As a user, I can search for flights with flexible dates.

...as "n days between x and y."
 ...as "a weekend in December."
 ...as "± n days of x and y."

Major Effort

As a user, I can pay for my flight with VISA, MasterCard, Diners Club, or American Express.

...I can pay with one credit card type (of VISA, MC, DC, AMEX).
 ...I can pay with all four credit card types (VISA, MC, DC, AMEX).

Simple/Complex

As a user, I can search for flights between two destinations.

...specifying a max number of stops.
 ...including nearby airports.
 ...using flexible dates.
 ...etc.

Variations in Data

As a content manager, I can create news stories.

...in English.
 ...in Japanese.
 ...in Arabic.
 ...etc.

Data Entry Methods

As a user, I can search for flights between two destinations.

...using simple date input.
 ...with a fancy calendar UI.

Defer Performance

As a user, I can search for flights between two destinations.

...(slow - just get it done, show a "searching" animation).
 ...(in under 5 seconds).

Operations (e.g. CRUD)

As a user, I can manage my account.

...I can sign up for an account.
 ...I can edit my account settings.
 ...I can cancel my account.

Break Out a Spike

As a user, I can pay by credit card.

Investigate credit card processing.
 Implement credit card processing (as one or more stories).

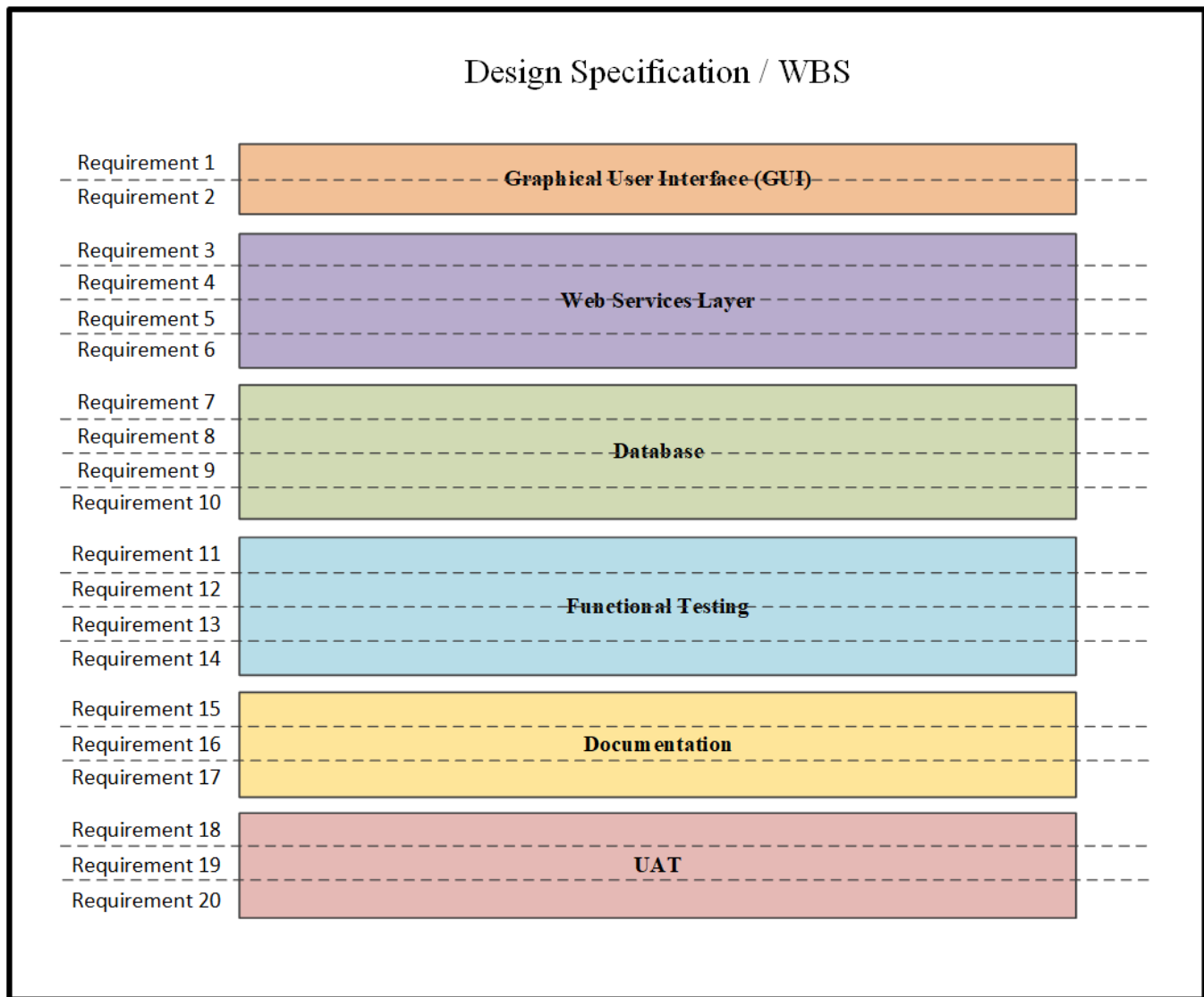
Copyright ©2009 Humanizing Work, LLC

CSPO-2.6 Explain at least two approaches to identify small, valuable, and usable Increments

Traditional Requirements Definition

Horizontal Components

Functional Teams



Agile Requirements Definition

Vertical Slices / Increments

Feature Teams

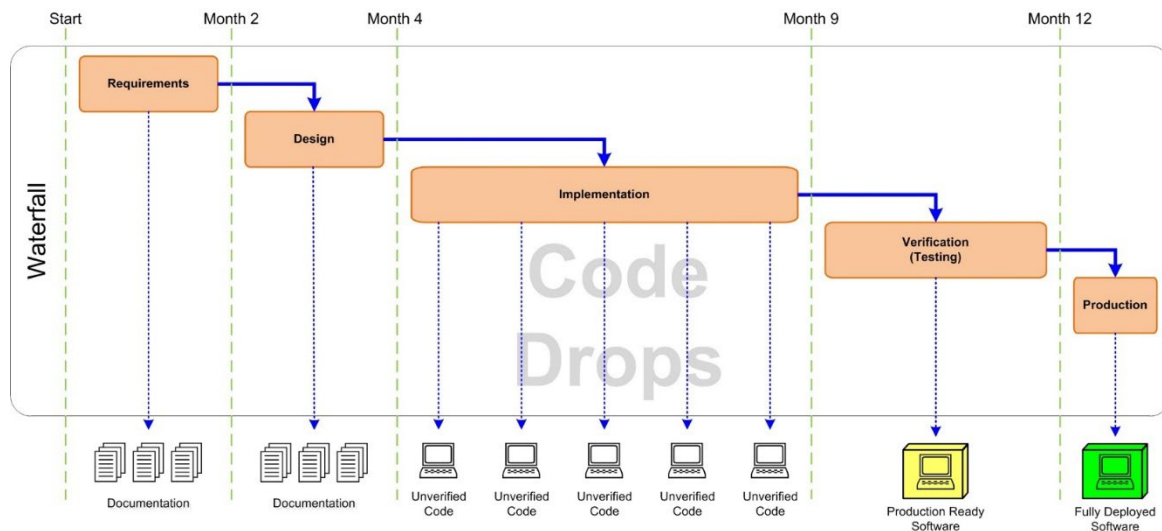
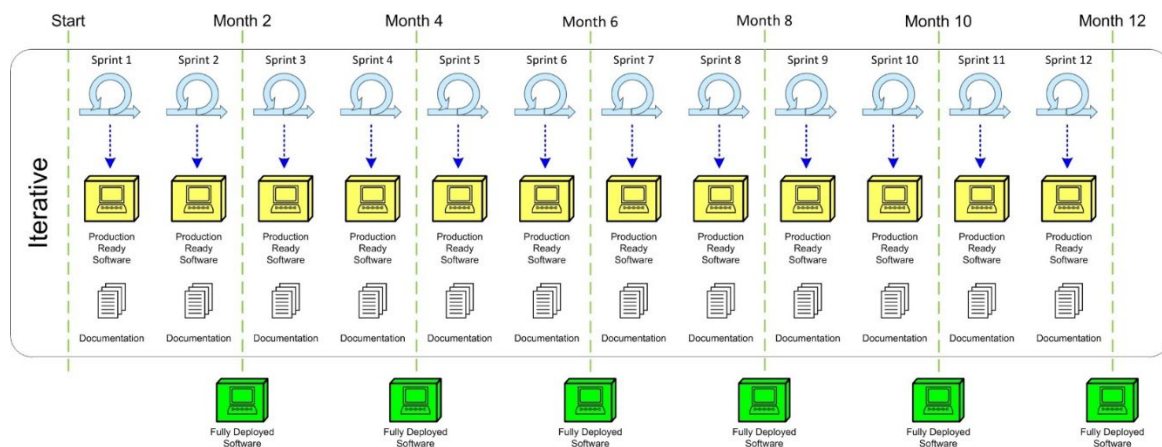
In order to deliver real value to customers as fully-functional features, Product Backlog items must be "cross-cutting" features which represent vertical slices through the architecture:

Product Backlog Items							
Epic 1			Epic 2		Epic 3		
Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Feature 8
			Graphical User Interface (GUI)				
			Web Services Layer				
			Database				
			Functional Testing				
			Documentation				
			UAT				

In this example, we see a web-based application with a middle-tier that integrates between an interface and a back-end database. The key is to create features which are decoupled or independent of each other as much as possible. The result is features that are actually usable from end to end, which is the only way value is truly delivered..

Validating Product Assumptions

CSPO-4.1 Describe how Scrum supports validating product assumptions

Example: 12 Month Waterfall Project**Example: 12 Month Scrum Project**

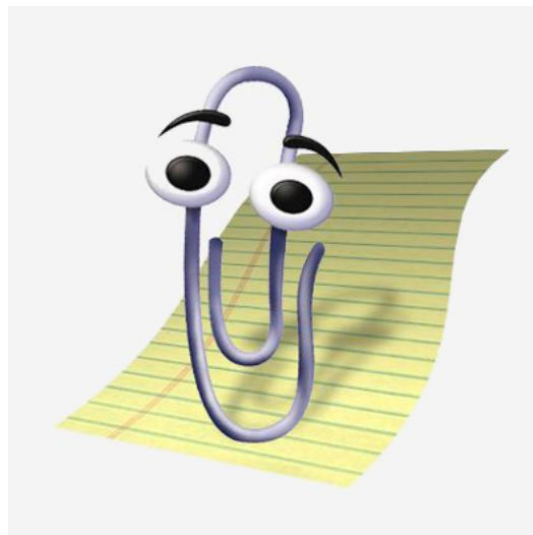
- Scrum reduces the risk of uncertainty by proving product assumptions
- Scrum accelerates the value delivery curve by giving value to the customer early and often

CSPO-4.2 Compare at least three approaches to validate product assumptions by their cost and quality of learning

A/B testing



Concierge MVP



• Wizard of Oz MVP

"Fake it 'til you make it."

E.g. Zappos, Amazon

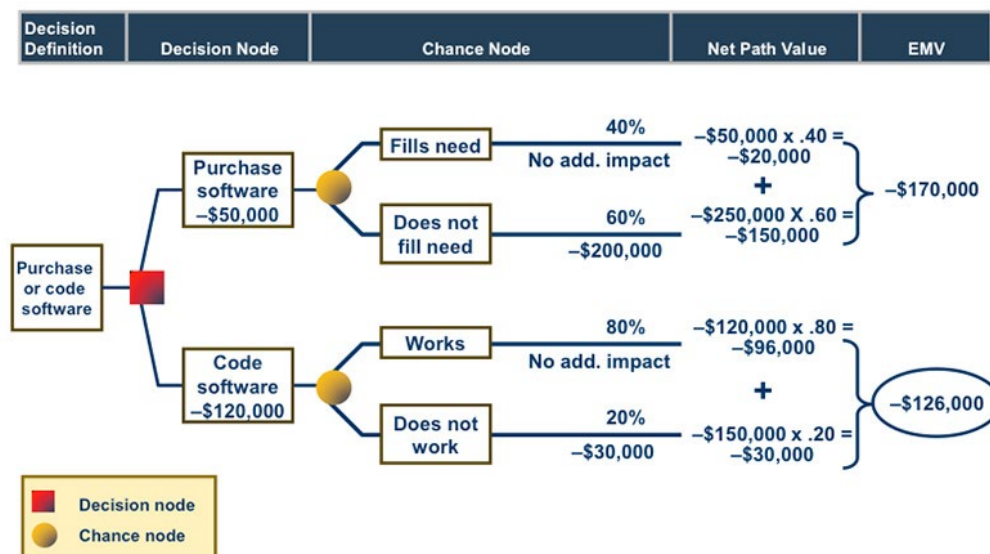


Functional prototypes

(Adobe XD, Axure RP, Balsamiq, Bubble, Figma, Frammer, Justinmind, Marvel, Miro, ProtoPie, Sketch, Webflow)

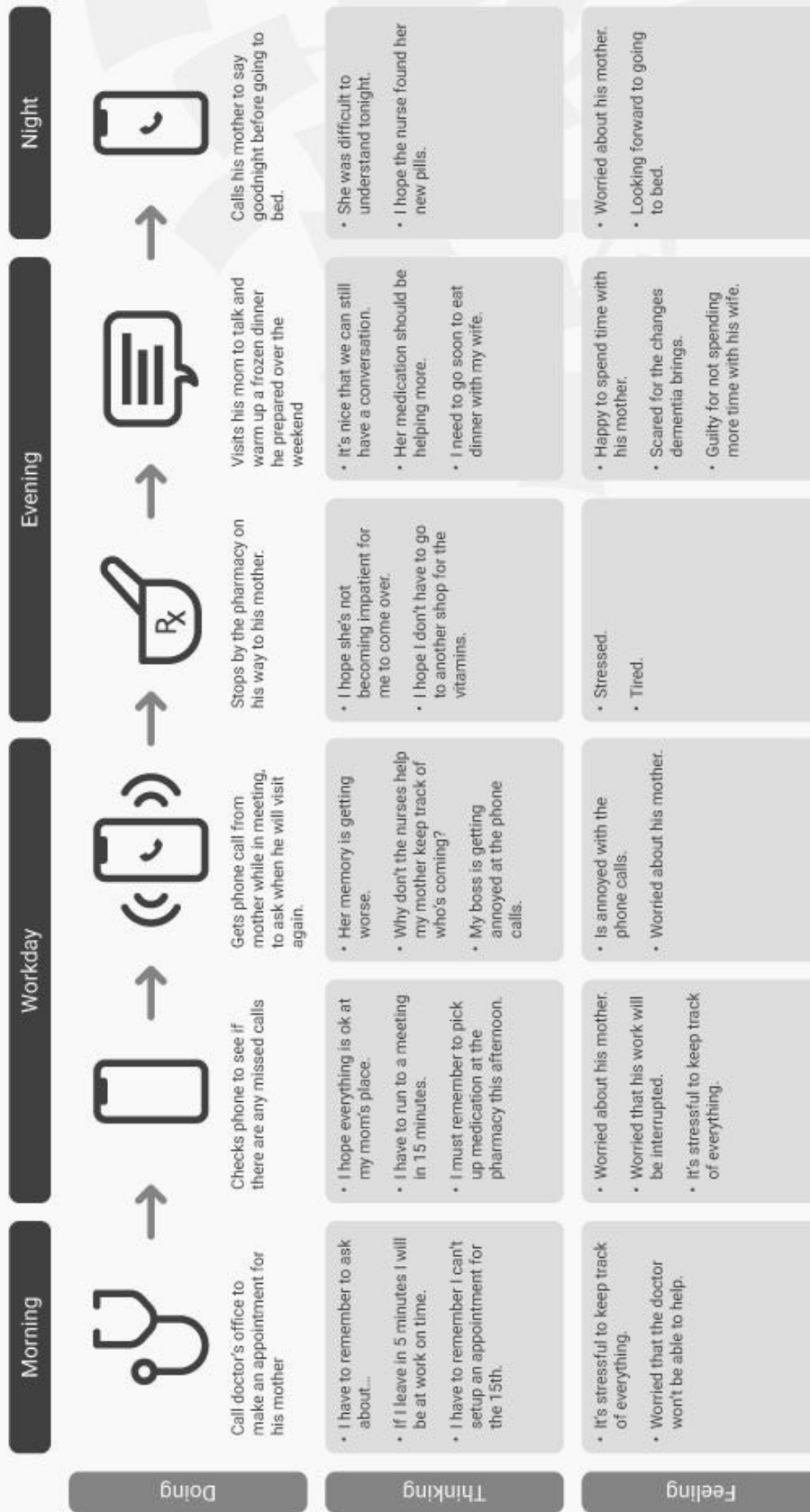


Decision Trees / Matrices



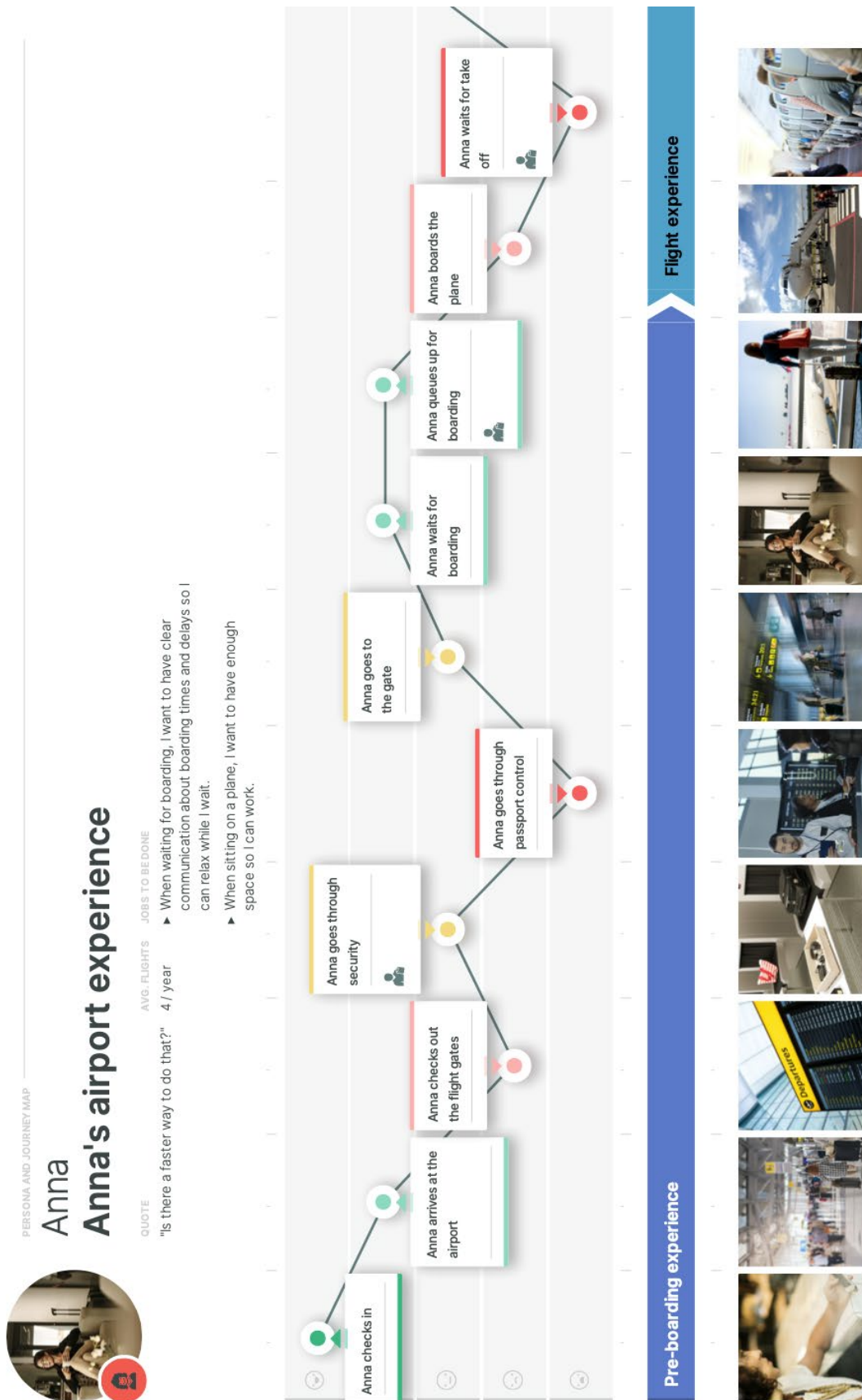
Customer Journey Maps

User Journey Map: Example



INTERACTION DESIGN
FOUNDATION

INTERACTION-DESIGN.ORG



The New Patient's Journey

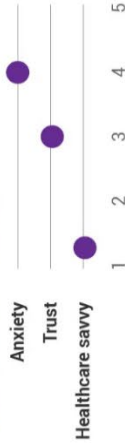
Opportunities to impact the experience at ABC Health



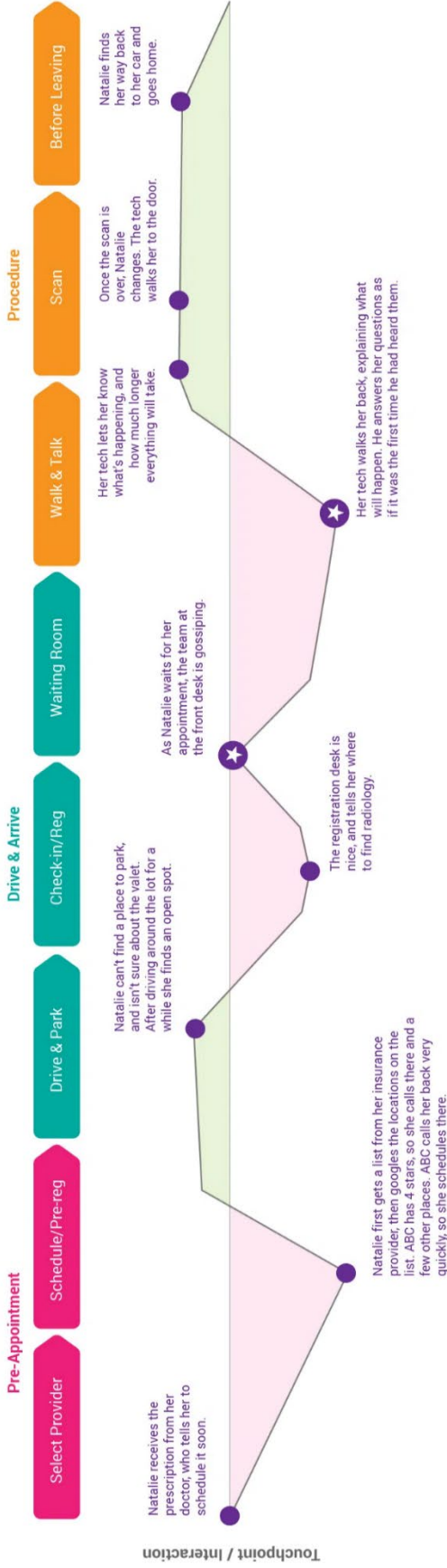
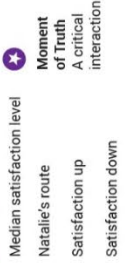
Newbie Natalie

Age 34
Background This is Natalie's first time for an MRI, so she's a bit anxious about it.
Insight Natalie's doctor's office printed out some WebMD pages about MRIs to help her understand what to expect.

Experience Factors



Legend



Thoughts

- "My doctor didn't help me find a place at all. They just gave me a script."
- "It seemed strange to me that only one hospital called me back. But they seemed really nice on the phone, and they were close to my home."
- "I've never been here before. Which entrance do I use?"
- "This place is in the basement?"
- "Wow, how unprofessional. Did I make the right decision coming here?"
- "I felt much better once I was with the tech. He was very professional and kind."
- "Every time I started getting nervous, he checked in and kept me encouraged."
- "Once the procedure was over, I guess it was just done."

ACTIVITY

For this activity we will be again be working in our cohort groups.

Each group will be responsible for creating TWO Customer Journey Maps:

As-is State

Define a particular experience or workflow that a customer is currently encountering that relates so the problem your product is intending to solve. Be sure to call out the pain points as opportunities to explore.

To-be State

For that same experience or workflow defined in the As-is State, describe what it will look like with your product or service solving the customer's pain points and making the experience AWESOME!

You will have **45 minutes** to work on these two journey maps.
When we come back together, your group should be ready to present.

Product Backlog Items

ACTIVITY

CSPO-5.6 Create a Product Backlog that supports the achievement of a Product Goal

CSPO-5.7 Create at least one Product Backlog item that includes a description of desired outcome and value

CSPO-5.8 Practice at least one approach to refining the Product Backlog

For this activity we will be again be working in our cohort groups.

Each group will be responsible for creating a Product Backlog for the first release in their product roadmap from yesterday.

You should practice using ALL the templates we mentioned previously and ensure that everything you create has acceptance criteria. You can also identify a taxonomy for classification with themes and other categories.

If you are not comfortable using the first release in your product roadmap, you may instead use the Journey Map that we just created and elaborate that into PBIs.

You will have **45 minutes** to work on these two journey maps.
When we come back together, your group should be ready to present.

Afterword



Dear student-

You just sat through my class and became certified.

I am very passionate about coaching and training. I go to great lengths and do everything in my power to bring a quality, enjoyable classroom experience to my students. Mostly, I try to have fun and spread the fun with everyone. In making these classes fun, I like to kid around and such. Hopefully you picked up on my sense of humor and sarcasm. If not, see the article on the next page from Seth Godin...

I also push the envelope and do not treat you with kid gloves. I love you too much as my students to sugarcoat the message for you. Agile is a HUGE paradigm shift in thinking. Improvement is a change. Don't expect to be Agile without making changes in your thinking.

Because of the diverse background and experience of the classroom participants, it is not always possible to discuss what is important to each person. If you need additional help, I am happy to provide whatever coaching I can, free of charge, beyond the classroom. So, please reach out and I will see what I can do for you. In some cases, I can answer your questions quickly via Email. In other cases, we can figure out what makes sense in terms of me coming to coach your organization.

Also, feedback is extremely important. It is a key driving principle of Agile and especially Scrum, as we have shown. We ask each student to complete a survey for the class in order to capture feedback on how we might improve and to provide one last opportunity to discuss things that were enjoyable or things that were a disappointment. Please take the time to let us know how we did. As it says in the Terms and Conditions, please reach out to me first before you talk to *(continued)* if you weren't happy. Let's understand each other. It's a key to not only Agile, but civilized society.

Even more candidly, if there was something you weren't happy with, have the decency to come to me first and discuss. I can't fix the situation if I don't know there is a problem.

In some very rare cases (only a handful of cases in >11 years), some people have complained to the Scrum Alliance about something that was essentially a customer satisfaction issue. That is completely disrespectful and contrary to the Scrum Values to do that. It tells me that the person didn't learn a thing about being Agile and should relinquish their certification...

Be human. Be real. Let's talk.

I wish you all the best in your Agile journey.

Peace and blessings.

-Daniel



*“Just be yourself, sir. Whatever happens,
they can't take that away from you.”*

For the **one person** who didn't get the joke...

The fabled comedian is killing it at a club that seats 400. One guy in the back, though, isn't laughing.

Miles Davis was shunned by a few people in the audience, even at his coolest.

The theater critic at the Times might not like this play, the one that made people cry and sold tickets for years.

And just about every blog post and book listing collects a trolling comment from someone who didn't like it, didn't read it or didn't agree with it (or all three) and isn't shy about speaking up with a sharp tongue.

For those people, the message from the creator of the work is clear:

"It's not for you."

Unanimity is impossible unless you are willing to be invisible. We can be unanimous in our lack of feedback for the invisible one.

For everyone else, though, the ability to say, "It's not for you," is the foundation for creating something brave and important. You can't do your best work if you're always trying to touch the untouchable, or entertain those that refuse to be entertained.

"It's not for you."

This is easy to say and incredibly difficult to do.

You don't have much choice, though, not if you want your work to matter.

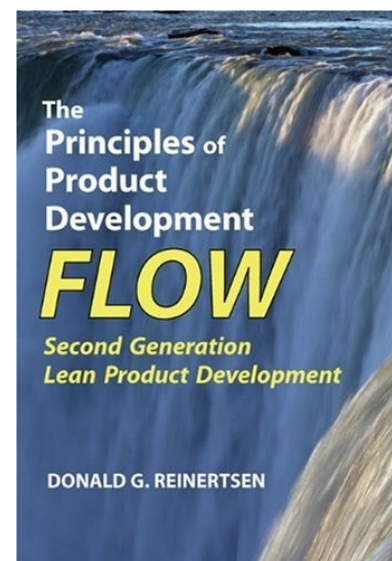
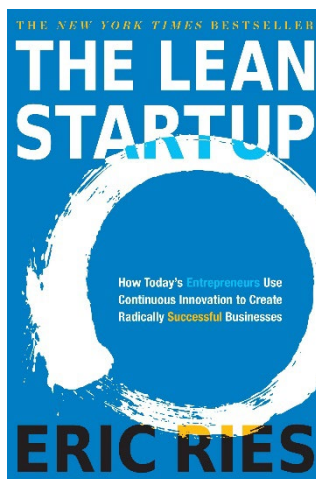
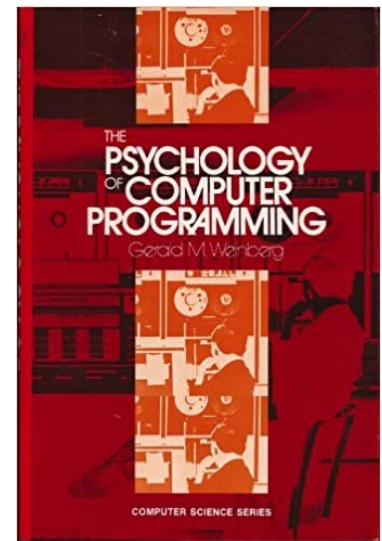
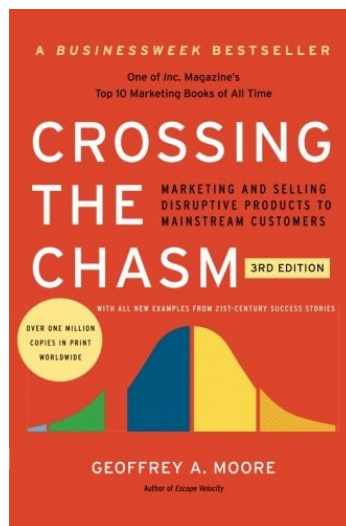
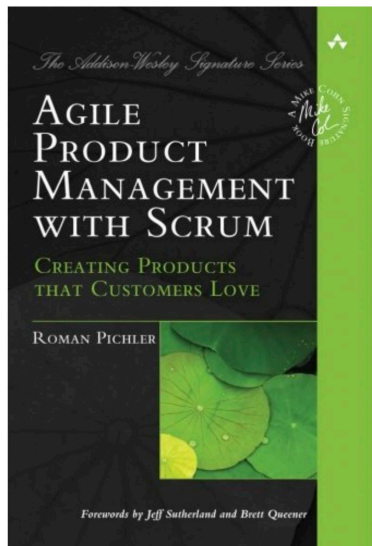
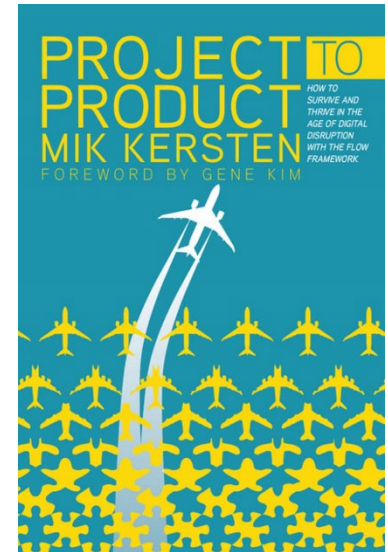
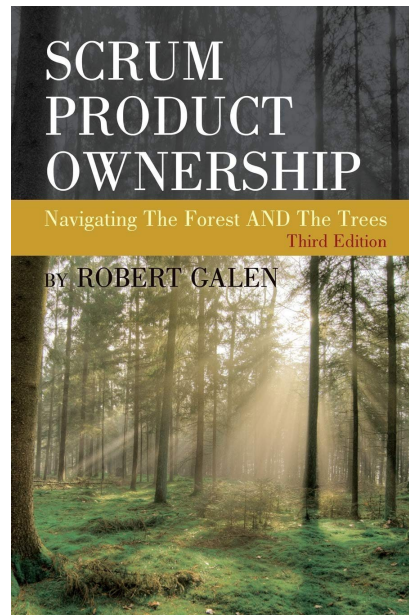
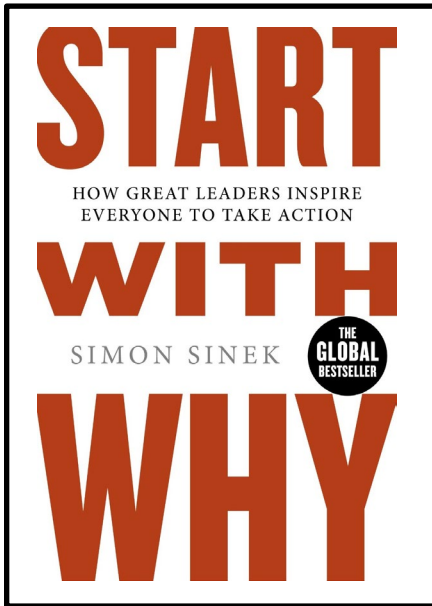
Seth Godin



What's Next?

1. **PLEASE... go back and RE-READ EVERYTHING AGAIN.** All of your questions have been answered already in this workbook and by Daniel in the class. Maybe watch the videos again on the KnowledgeHut site Prism. Download and read the Preclass Work PDF as well.
2. Daniel will check the Zoom connection logs to make sure you were in class the FULL time.
3. Assuming you were there the full time AND participating in the course with video camera turned on, etc. Daniel will contact KnowledgeHut to inform them that you can be uploaded to the Scrum Alliance system. At this point, if you need help or support, you must contact KnowledgeHut.
4. You will receive an Email from Scrum Alliance with additional instructions in it. Read these instructions very carefully. If you do NOT receive an Email from Scrum Alliance within 3 days of the course, you must contact KnowledgeHut.
5. Once you do receive the Email from Scrum Alliance, if you need help or support you must contact Scrum Alliance. Neither Daniel nor KnowledgeHut can do anything for you at this point.
6. In general, Scrum Alliance will ask you to create a profile by entering additional personal information.
7. You will have a dashboard in your profile, which is where you manage your certification. Your certificate is here. Your digital badge and logos are here. You log your SEUs here.
8. You will accept your CSPO license agreement.
9. You will go out and be an awesome Product Owner in the world as a newly certified CSPO.
10. Read this article:
Getting Hired With Your New Certification
<https://resources.scrumalliance.org/Article/put-your-new-scrummaster-skills-to-work>
11. I don't do resume reviews. I also don't have any magical words of wisdom when it comes to finding your first job as a CSPO other than this: BUILD A PERSONAL BRAND ONLINE
If you put in the sweat equity, you will eventually be successful. It may take time. Have faith.

Must-Read Books on Product Ownership



Appendix

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Purpose of the Scrum Guide

We developed Scrum in the early 1990s. We wrote the first version of the Scrum Guide in 2010 to help people worldwide understand Scrum. We have evolved the Guide since then through small, functional updates. Together, we stand behind it.

The Scrum Guide contains the definition of Scrum. Each element of the framework serves a specific purpose that is essential to the overall value and results realized with Scrum. Changing the core design or ideas of Scrum, leaving out elements, or not following the rules of Scrum, covers up problems and limits the benefits of Scrum, potentially even rendering it useless.

We follow the growing use of Scrum within an ever-growing complex world. We are humbled to see Scrum being adopted in many domains holding essentially complex work, beyond software product development where Scrum has its roots. As Scrum's use spreads, developers, researchers, analysts, scientists, and other specialists do the work. We use the word “developers” in Scrum not to exclude, but to simplify. If you get value from Scrum, consider yourself included.

As Scrum is being used, patterns, processes, and insights that fit the Scrum framework as described in this document, may be found, applied and devised. Their description is beyond the purpose of the Scrum Guide because they are context sensitive and differ widely between Scrum uses. Such tactics for using within the Scrum framework vary widely and are described elsewhere.

Ken Schwaber & Jeff Sutherland November 2020

© 2020 Ken Schwaber and Jeff Sutherland

This publication is offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <https://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <https://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this Scrum Guide, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

Purpose of the Scrum Guide	1
Scrum Definition	3
Scrum Theory	3
Transparency	3
Inspection.....	4
Adaptation	4
Scrum Values.....	4
Scrum Team.....	5
Developers	5
Product Owner	5
Scrum Master.....	6
Scrum Events	7
The Sprint	7
Sprint Planning	8
Daily Scrum.....	9
Sprint Review	9
Sprint Retrospective	10
Scrum Artifacts.....	10
Product Backlog	10
Commitment: Product Goal.....	11
Sprint Backlog	11
Commitment: Sprint Goal.....	11
Increment	11
Commitment: Definition of Done.....	12
End Note	13
Acknowledgements	13
People	13
Scrum Guide History	13

Scrum Definition

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

1. A Product Owner orders the work for a complex problem into a Product Backlog.
2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
4. *Repeat*

Scrum is simple. Try it as is and determine if its philosophy, theory, and structure help to achieve goals and create value. **The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory. Scrum is built upon by the collective intelligence of the people using it. Rather than provide people with detailed instructions, the rules of Scrum guide their relationships and interactions.**

Various processes, techniques and methods can be employed within the framework. Scrum wraps around existing practices or renders them unnecessary. Scrum makes visible the relative efficacy of current management, environment, and work techniques, so that improvements can be made.

Scrum Theory

Scrum is founded on empiricism and lean thinking. Empiricism asserts that knowledge comes from experience and making decisions based on what is observed. Lean thinking reduces waste and focuses on the essentials.

Scrum employs an iterative, incremental approach to optimize predictability and to control risk. Scrum engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed.

Scrum combines four formal events for inspection and adaptation within a containing event, the Sprint. These events work because they implement the empirical Scrum pillars of transparency, inspection, and adaptation.

Transparency

The emergent process and work must be visible to those performing the work as well as those receiving the work. With Scrum, important decisions are based on the perceived state of its three formal artifacts. Artifacts that have low transparency can lead to decisions that diminish value and increase risk.

Transparency enables inspection. Inspection without transparency is misleading and wasteful.

Inspection

The Scrum artifacts and the progress toward agreed goals must be inspected frequently and diligently to detect potentially undesirable variances or problems. To help with inspection, Scrum provides cadence in the form of its five events.

Inspection enables adaptation. Inspection without adaptation is considered pointless. Scrum events are designed to provoke change.

Adaptation

If any aspects of a process deviate outside acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced must be adjusted. The adjustment must be made as soon as possible to minimize further deviation.

Adaptation becomes more difficult when the people involved are not empowered or self-managing. A Scrum Team is expected to adapt the moment it learns anything new through inspection.

Scrum Values

Successful use of Scrum depends on people becoming more proficient in living five values:

Commitment, Focus, Openness, Respect, and Courage

The Scrum Team commits to achieving its goals and to supporting each other. Their primary focus is on the work of the Sprint to make the best possible progress toward these goals. The Scrum Team and its stakeholders are open about the work and the challenges. Scrum Team members respect each other to be capable, independent people, and are respected as such by the people with whom they work. The Scrum Team members have the courage to do the right thing, to work on tough problems.

These values give direction to the Scrum Team with regard to their work, actions, and behavior. The decisions that are made, the steps taken, and the way Scrum is used should reinforce these values, not diminish or undermine them. The Scrum Team members learn and explore the values as they work with the Scrum events and artifacts. When these values are embodied by the Scrum Team and the people they work with, the empirical Scrum pillars of transparency, inspection, and adaptation come to life building trust.

Scrum Team

The fundamental unit of Scrum is a small team of people, a Scrum Team. The Scrum Team consists of one Scrum Master, one Product Owner, and Developers. Within a Scrum Team, there are no sub-teams or hierarchies. It is a cohesive unit of professionals focused on one objective at a time, the Product Goal.

Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint. They are also self-managing, meaning they internally decide who does what, when, and how.

The Scrum Team is small enough to remain nimble and large enough to complete significant work within a Sprint, typically 10 or fewer people. In general, we have found that smaller teams communicate better and are more productive. If Scrum Teams become too large, they should consider reorganizing into multiple cohesive Scrum Teams, each focused on the same product. Therefore, they should share the same Product Goal, Product Backlog, and Product Owner.

The Scrum Team is responsible for all product-related activities from stakeholder collaboration, verification, maintenance, operation, experimentation, research and development, and anything else that might be required. They are structured and empowered by the organization to manage their own work. Working in Sprints at a sustainable pace improves the Scrum Team's focus and consistency.

The entire Scrum Team is accountable for creating a valuable, useful Increment every Sprint. Scrum defines three specific accountabilities within the Scrum Team: the Developers, the Product Owner, and the Scrum Master.

Developers

Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint.

The specific skills needed by the Developers are often broad and will vary with the domain of work. However, the Developers are always accountable for:

- Creating a plan for the Sprint, the Sprint Backlog;
- Instilling quality by adhering to a Definition of Done;
- Adapting their plan each day toward the Sprint Goal; and,
- Holding each other accountable as professionals.

Product Owner

The Product Owner is accountable for maximizing the value of the product resulting from the work of the Scrum Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

The Product Owner is also accountable for effective Product Backlog management, which includes:

- Developing and explicitly communicating the Product Goal;
- Creating and clearly communicating Product Backlog items;
- Ordering Product Backlog items; and,
- Ensuring that the Product Backlog is transparent, visible and understood.

The Product Owner may do the above work or may delegate the responsibility to others. Regardless, the Product Owner remains accountable.

For Product Owners to succeed, the entire organization must respect their decisions. These decisions are visible in the content and ordering of the Product Backlog, and through the inspectable Increment at the Sprint Review.

The Product Owner is one person, not a committee. The Product Owner may represent the needs of many stakeholders in the Product Backlog. Those wanting to change the Product Backlog can do so by trying to convince the Product Owner.

Scrum Master

The Scrum Master is accountable for establishing Scrum as defined in the Scrum Guide. They do this by helping everyone understand Scrum theory and practice, both within the Scrum Team and the organization.

The Scrum Master is accountable for the Scrum Team's effectiveness. They do this by enabling the Scrum Team to improve its practices, within the Scrum framework.

Scrum Masters are true leaders who serve the Scrum Team and the larger organization. The

Scrum Master serves the Scrum Team in several ways, including:

- Coaching the team members in self-management and cross-functionality;
- Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
- Causing the removal of impediments to the Scrum Team's progress; and,
- Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.

The Scrum Master serves the Product Owner in several ways, including:

- Helping find techniques for effective Product Goal definition and Product Backlog management;
- Helping the Scrum Team understand the need for clear and concise Product Backlog items;
- Helping establish empirical product planning for a complex environment; and,
- Facilitating stakeholder collaboration as requested or needed.

The Scrum Master serves the organization in several ways, including:

- Leading, training, and coaching the organization in its Scrum adoption;
- Planning and advising Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
- Removing barriers between stakeholders and Scrum Teams.

Scrum Events

The Sprint is a container for all other events. Each event in Scrum is a formal opportunity to inspect and adapt Scrum artifacts. These events are specifically designed to enable the transparency required.

Failure to operate any events as prescribed results in lost opportunities to inspect and adapt. Events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum.

Optimally, all events are held at the same time and place to reduce complexity.

The Sprint

Sprints are the heartbeat of Scrum, where ideas are turned into value.

They are fixed length events of one month or less to create consistency. A new Sprint starts immediately after the conclusion of the previous Sprint.

All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.

During the Sprint:

- No changes are made that would endanger the Sprint Goal;
- Quality does not decrease;
- The Product Backlog is refined as needed; and,
- Scope may be clarified and renegotiated with the Product Owner as more is learned.

Sprints enable predictability by ensuring inspection and adaptation of progress toward a Product Goal at least every calendar month. When a Sprint's horizon is too long the Sprint Goal may become invalid, complexity may rise, and risk may increase. Shorter Sprints can be employed to generate more learning

cycles and limit risk of cost and effort to a smaller time frame. Each Sprint may be considered a short project.

Various practices exist to forecast progress, like burn-downs, burn-ups, or cumulative flows. While proven useful, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision making.

A Sprint could be cancelled if the Sprint Goal becomes obsolete. Only the Product Owner has the authority to cancel the Sprint.

Sprint Planning

Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint. This resulting plan is created by the collaborative work of the entire Scrum Team.

The Product Owner ensures that attendees are prepared to discuss the most important Product Backlog items and how they map to the Product Goal. The Scrum Team may also invite other people to attend Sprint Planning to provide advice.

Sprint Planning addresses the following topics:

Topic One: Why is this Sprint valuable?

The Product Owner proposes how the product could increase its value and utility in the current Sprint. The whole Scrum Team then collaborates to define a Sprint Goal that communicates why the Sprint is valuable to stakeholders. The Sprint Goal must be finalized prior to the end of Sprint Planning.

Topic Two: What can be Done this Sprint?

Through discussion with the Product Owner, the Developers select items from the Product Backlog to include in the current Sprint. The Scrum Team may refine these items during this process, which increases understanding and confidence.

Selecting how much can be completed within a Sprint may be challenging. However, the more the Developers know about their past performance, their upcoming capacity, and their Definition of Done, the more confident they will be in their Sprint forecasts.

Topic Three: How will the chosen work get done?

For each selected Product Backlog item, the Developers plan the work necessary to create an Increment that meets the Definition of Done. This is often done by decomposing Product Backlog items into smaller work items of one day or less. How this is done is at the sole discretion of the Developers. No one else tells them how to turn Product Backlog items into Increments of value.

The Sprint Goal, the Product Backlog items selected for the Sprint, plus the plan for delivering them are together referred to as the Sprint Backlog.

Sprint Planning is timeboxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Daily Scrum

The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.

The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, it is held at the same time and place every working day of the Sprint. If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.

The Developers can select whatever structure and techniques they want, as long as their Daily Scrum focuses on progress toward the Sprint Goal and produces an actionable plan for the next day of work. This creates focus and improves self-management.

Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings.

The Daily Scrum is not the only time Developers are allowed to adjust their plan. They often meet throughout the day for more detailed discussions about adapting or re-planning the rest of the Sprint's work.

Sprint Review

The purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations. The Scrum Team presents the results of their work to key stakeholders and progress toward the Product Goal is discussed.

During the event, the Scrum Team and stakeholders review what was accomplished in the Sprint and what has changed in their environment. Based on this information, attendees collaborate on what to do next. The Product Backlog may also be adjusted to meet new opportunities. The Sprint Review is a working session and the Scrum Team should avoid limiting it to a presentation.

The Sprint Review is the second to last event of the Sprint and is timeboxed to a maximum of four hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Sprint Retrospective

The purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness.

The Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done. Inspected elements often vary with the domain of work.

Assumptions that led them astray are identified and their origins explored. The Scrum Team discusses what went well during the Sprint, what problems it encountered, and how those problems were (or were not) solved.

The Scrum Team identifies the most helpful changes to improve its effectiveness. The most impactful improvements are addressed as soon as possible. They may even be added to the Sprint Backlog for the next Sprint.

The Sprint Retrospective concludes the Sprint. It is timeboxed to a maximum of three hours for a one- month Sprint. For shorter Sprints, the event is usually shorter.

Scrum Artifacts

Scrum's artifacts represent work or value. They are designed to maximize transparency of key information. Thus, everyone inspecting them has the same basis for adaptation.

Each artifact contains a commitment to ensure it provides information that enhances transparency and focus against which progress can be measured:

- For the Product Backlog it is the Product Goal.
- For the Sprint Backlog it is the Sprint Goal.
- For the Increment it is the Definition of Done.

These commitments exist to reinforce empiricism and the Scrum values for the Scrum Team and their stakeholders.

Product Backlog

The Product Backlog is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum Team.

Product Backlog items that can be Done by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event. They usually acquire this degree of transparency after refining activities. Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items. This is an ongoing activity to add details, such as a description, order, and size. Attributes often vary with the domain of work.

The Developers who will be doing the work are responsible for the sizing. The Product Owner may influence the Developers by helping them understand and select trade-offs.

Commitment: Product Goal

The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against. The Product Goal is in the Product Backlog. The rest of the Product Backlog emerges to define “what” will fulfill the Product Goal.

A product is a vehicle to deliver value. It has a clear boundary, known stakeholders, well-defined users or customers. A product could be a service, a physical product, or something more abstract.

The Product Goal is the long-term objective for the Scrum Team. They must fulfill (or abandon) one objective before taking on the next.

Sprint Backlog

The Sprint Backlog is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), as well as an actionable plan for delivering the Increment (how).

The Sprint Backlog is a plan by and for the Developers. It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal.

Consequently, the Sprint Backlog is updated throughout the Sprint as more is learned. It should have enough detail that they can inspect their progress in the Daily Scrum.

Commitment: Sprint Goal

The Sprint Goal is the single objective for the Sprint. Although the Sprint Goal is a commitment by the Developers, it provides flexibility in terms of the exact work needed to achieve it. The Sprint Goal also creates coherence and focus, encouraging the Scrum Team to work together rather than on separate initiatives.

The Sprint Goal is created during the Sprint Planning event and then added to the Sprint Backlog. As the Developers work during the Sprint, they keep the Sprint Goal in mind. If the work turns out to be different than they expected, they collaborate with the Product Owner to negotiate the scope of the Sprint Backlog within the Sprint without affecting the Sprint Goal.

Increment

An Increment is a concrete stepping stone toward the Product Goal. Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together. In order to provide value, the Increment must be usable.

Multiple Increments may be created within a Sprint. The sum of the Increments is presented at the Sprint Review thus supporting empiricism. However, an Increment may be delivered to stakeholders prior to the end of the Sprint. The Sprint Review should never be considered a gate to releasing value.

Work cannot be considered part of an Increment unless it meets the Definition of Done.

Commitment: Definition of Done

The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.

The moment a Product Backlog item meets the Definition of Done, an Increment is born.

The Definition of Done creates transparency by providing everyone a shared understanding of what work was completed as part of the Increment. If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Sprint Review. Instead, it returns to the Product Backlog for future consideration.

If the Definition of Done for an increment is part of the standards of the organization, all Scrum Teams must follow it as a minimum. If it is not an organizational standard, the Scrum Team must create a Definition of Done appropriate for the product.

The Developers are required to conform to the Definition of Done. If there are multiple Scrum Teams working together on a product, they must mutually define and comply with the same Definition of Done.

End Note

Scrum is free and offered in this Guide. The Scrum framework, as outlined herein, is immutable. While implementing only parts of Scrum is possible, the result is not Scrum. Scrum exists only in its entirety and functions well as a container for other techniques, methodologies, and practices.

Acknowledgements

People

Of the thousands of people who have contributed to Scrum, we should single out those who were instrumental at the start: Jeff Sutherland worked with Jeff McKenna and John Scumniotales, and Ken Schwaber worked with Mike Smith and Chris Martin, and all of them worked together. Many others contributed in the ensuing years and without their help Scrum would not be refined as it is today.

Scrum Guide History

Ken Schwaber and Jeff Sutherland first co-presented Scrum at the OOPSLA Conference in 1995. It essentially documented the learning that Ken and Jeff gained over the previous few years and made public the first formal definition of Scrum.

The Scrum Guide documents Scrum as developed, evolved, and sustained for 30-plus years by Jeff Sutherland and Ken Schwaber. Other sources provide patterns, processes, and insights that complement the Scrum framework. These may increase productivity, value, creativity, and satisfaction with the results.

The complete history of Scrum is described elsewhere. To honor the first places where it was tried and proven, we recognize Individual Inc., Newspaper, Fidelity Investments, and IDX (now GE Medical).

© 2020 Ken Schwaber and Jeff Sutherland

This publication is offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <https://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <https://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this Scrum Guide, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

Bibliography

The books listed have had a deep and profound impact on my growth over the years as a coach, trainer, business owner, etc. I have drawn heavily upon these and others. I have done my best to provide a comprehensive list. If you get finished with these, there are hundreds of other books to continue on in your learning journey.

Adkins, Lyssa. 2010. *Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition*. Upper Saddle River, NJ: Addison-Wesley.

Anderson, David J. *Kanban: 2010. Successful Evolutionary Change for Your Technology Business*. Seattle: Blue Hole Press

Appelo, Jurgen. 2011. *Management 3.0: Leading Agile Developers, Developing Agile Leaders*. Upper Saddle River, NJ: Addison-Wesley.

Blank, S., Dorf, B. (2020). *The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company*. United Kingdom: Wiley.

Brooks, Frederick P. 1995. *The Mythical Man Month*. Reading, Mass: Addison-Wesley.

Cagan, M. (2017). *Inspired: How to create tech products customers love*. John Wiley & Sons.

Cockburn, Alistair. 2000. *Writing Effective Use Cases*. New Jersey: Addison-Wesley.

Cohn, Mike. 2004. *User Stories Applied: For Agile Software Development*. Boston: Addison-Wesley.

Cohn, Mike. 2006. *Agile Estimating and Planning*. Upper Saddle River NJ: Prentice Hall PTR.

Coplien, James O. 2005. *Organizational Patterns of Agile Software Development*. Upper Saddle River, NJ: Prentice Hall.

Denning, Stephen. 2010. *The Leader's Guide to Radical Management: Reinventing the Workplace for the 21st Century*. San Francisco: Jossey-Bass.

Derby, Esther and Larsen, Diana. 2006. *Agile Retrospectives: Making Good Teams Great*. Frisco, TX: Pragmatic Bookshelf

Foer, Joshua. 2011. *Moonwalking with Einstein: The Art and Science of Remembering Everything*. London: Penguin Books, Ltd.

Goleman, Daniel. 2006. *Emotional Intelligence: Why It Can Matter More Than IQ*. New York: Random House

Greenleaf, Robert K. 2002. *Servant Leadership: A Journey into The Nature of Legitimate Power and Greatness*. New Jersey: Paulist Press.

Gullo, Daniel J. 2016. *Real World Agility: Practical Guidance for Agile Practitioners*. Boston: Pearson Technical Group.

Highsmith, Jim. 2010. *Agile Project Management: Creating Innovative Products*. Upper Saddle River, NJ: Addison-Wesley.

Jeffries, Ron and Ann Anderson, Chet Hendrickson. 2000. *eXtreme Programming Installed*. Boston: Addison-Wesley.

Joiner, Bill and Stephen Josephs. 2007. *Leadership Agility: Five Levels of Mastery for Anticipating and Initiating Change*. San Francisco: Jossey-Bass.

- Kerth, Norman. 2001. *Project Retrospectives: A Handbook for Team Reviews*. New York: Dorsett House
- Kleon, Austin. 2012. *Steal Like an Artist: 10 Things Nobody Told You About Being Creative*. New York: Workman Publishing Company Inc.
- Krebs, Jochen. 2009. *Agile Portfolio Management*. Redmond, WA: Microsoft Press.
- Kohn, Alfie. 1999. *Punished by Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes*. Boston: Houghton Mifflin.
- Kurzweil, Ray. 2012. *How to Create a Mind: The Secret of Human Thought Revealed*. New York: Penguin Books Ltd.
- Larman, Craig. 2009. *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. Upper Saddle River, NJ: Addison-Wesley.
- Lasater, Judith Hanson and Lasater, Ike K. 2009. *What we Say Matters: Practicing Non-Violent Communication*. Berkeley: Rodmell Press.
- Lencioni, Patrick. 2002. *The Five Dysfunctions of a Team: A Leadership Fable*. San Francisco: Jossey-Bass.
- Leffingwell, Dean. 2011. *Agile Software Requirements: Lean Requirements Practices For Teams, Programs, and the Enterprise*. Upper Saddle River, NJ: Addison-Wesley.
- Martinez, Sylvia Libow and Stager, Gary S. 2013. *Invent To Learn: Making, Tinkering, and Engineering in the Classroom*. Torrance, CA: Constructing Modern Knowledge Press.
- McCarthy, Jim. 2002. *Software for Your Head: Core Protocols for Creating and Maintaining Shared Vision*. Boston: Addison-Wesley.
- McConnell, Steven C. 2004. *Code Complete: A Practical Handbook Of Software Construction*. Redmond, WA: Microsoft Press.
- Meadows, Donella H. 2008. *Thinking in Systems: A Primer*. Vermont: Chelsea Green Publishing Company
- Mezick, Daniel. 2012. *The Culture Game: Tools for the Agile Manager*. Hartford: Free Standing Press.
- Moore, Geoffrey A. 2002. *Crossing The Chasm*. New York: HarperCollins Publishers.
- Olsen, D. (2015). *The Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback*. Switzerland: Wiley.
- Oshry, B. (1994). *In the Middle*. United States: Power & Systems.
- Owen, Harrison. 2008. *Open Space Technology: A User's Guide*. San Francisco: Berrett-Koehler Publishers, Inc.
- Perkins, Dennis N. T. 2000. *Leading at the Edge : Leadership Lessons from the Extraordinary Saga of Shackleton's Antarctic Expedition*. New York: AMACOM

- Pichler, Roman. 2010. *Agile Product Management with Scrum: Creating Products that Customers Love*. Upper Saddle River, NJ: Addison-Wesley.
- Pink, Daniel. 2011. *Drive: The Surprising Truth About What Motivates Us*. New York: Penguin Group.
- Poppendieck, M., & Poppendieck, T. (2003). *Lean software development: an agile toolkit*. Addison-Wesley.
- Reinertsen, Donald G. 2009. *The Principles of Product Development FLOW: Second Generation Lean Product Development*. Redondo Beach, CA: Celeritas Publishing
- Ries, Eric. 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York: Crown Business.
- Robinson, Sir Ken. 2001. *Out of Our Minds: Learning to Be Creative*. West Sussex, United Kingdom: Capstone Publishing Ltd.
- Rock, David and Linda J. Page. 2009. *Coaching with the Brain in Mind: Foundations for Practice*. Hoboken, NJ: John Wiley & Sons, Inc.
- Rosenberg, Marshall. 2003. *Non-Violent Communication: A Language of Life*. Encinitas, CA: PuddleDancer Press.
- Satir, Virginia and Banmen, John and Gerber, Jane and Gomori, Maria. 1991. *The Satir Model: Family Therapy and Beyond*. Palo Alto, CA: Science and Behavior Books Inc.
- Schwaber, Ken and Mike Beedle. 2002. *Agile Software Development with Scrum*. Upper Saddle River, NJ: Prentice Hall.
- Scott, Susan. 2011. *Fierce Conversations: Achieving Success at Work and in Life, One Conversation at a Time*. New York: The Berkley Publishing Group.
- Seddon, John. 2005. *Freedom from Command and Control: Rethinking Management for Lean Service*. New York: Productivity Press.
- Senge, Peter. 2006. *The Fifth Discipline*. New York: Currency – Doubleday.
- Shalloway, Alan and Beaver, Guy and Trott, James R. 2010. *Lean-Agile Software Development: Achieving Enterprise Agility*. Upper Saddle River, NJ: Addison-Wesley.
- Sheridan, Richard. 2013. *Joy, Inc. How We Built a Workplace People Love*. New York: Penguin Group
- Silsbee, Doug. 2008. *Presence-Based Coaching: Cultivating Self-Generative Leaders Through Mind, Body, and Heart*. San Francisco: Jossey-Bass.
- Sinek, Simon. 2009. *Start with Why: How Great Leaders Inspire Everyone to Take Action*. New York: Penguin Group.
- Sliger, Michele and Broderick, Stacia. 2008. *The Software Project Manager's Bridge to Agility*. Boston: Addison-Wesley.
- Sutherland, Jeff. 2014. *Scrum: The Art of Doing Twice the Work in Half the Time*. New York: Crown Business.
- Taleb, Nassim Nicholas. 2012. *Antifragile: Things That Gain from Disorder*. New York: Random House Publishing Group.

Weinberg, Gerald. 1998. *The Psychology of Computer Programming*. New York: Dorset House Publishing.

Weinberg, Gerald and Satir, Virginia. 1986. *The Secrets of Consulting*. New York: Dorset House Publishing

Wheatley, Margaret. 2006. *Leadership and the New Science: Discovering Order in a Chaotic World*. Oakland, CA:

Berrett-Koehler Publishers Inc.

Willink, Jocko and Babin, Leif. 2015. *Extreme Ownership: How U.S. Navy Seals Lead and Win*. New York: St. Martin's Press

Culturama

Various TV, Movie, and Music References

Being a visual and auditory person (and someone who enjoys laughing), I like to use a lot of imagery and analogies from movies, TV, music, etc. I don't make all of these references EVERY class, but in case you heard me reference something and were wondering "Huh?? What did he say?" here are some rough visuals that help me remember Agile concepts.

Star Wars: Episode IV – Being transparent vs. pretending everything is ok. A commander in the Death Star calls down to the detention block where Luke and Han have just taken control. Han says "We're all fine down here... how are you?"

Brazil – This whole movie is a classic parody of futuristic corporate life and bureaucracy. In particular, there is Harry Tuttle, the renegade HVAC repair man who says he left the Ministry because he couldn't stand the paperwork. "This whole place could be on fire and I couldn't turn a screwdriver without a 27B/6." This is the classic "comprehensive documentation" and "process and tools".

The Office (US) – Trusting technology too much and mindlessly following directions can have catastrophic consequences. Michael and Dwight found this out when Michael followed the GPS without using really thinking and drove right into a pond.

Spinal Tap – There is no "Sprint 0". Just like, it doesn't make sense to have amps that "go to 11".

Army of Darkness

Office Space

Columbo – Just one more question...

Poirot – The little grey cells are working overtime.

Mosquito Coast – In trying to escape technology and the trappings of modern culture, they bring these things with them and prove that man is inherently evil.

Trading Places – "Just be yourself, sir. No matter what happens, they can't take that away from you."