aqqurite

Certified Scrum Product Owner

Arne Åhlander

V 2.5

Contacts

Arne Åhlander
arne.ahlander@aqqurite.se
www.aqqurite.se
www.twitter.com/ArneAhl
http://www.linkedin.com/in/arneahlander

aqqurite

Agenda

Day 1

Intro
The Scrum Framework
The Role of the Product Owner

Purpose and Strategy
Product Vision
Product Discovery

Day 2

Roadmaps and Release Planning
Product Assumptions
Product Backlog
- Outcome and Output

- Value & Ordering
- Creating & Refinin
Closing questions
Summary

Workbook Structure

Part 1 - Need to know

- The core of the course
- This part covers the Learning Objectives

Part 2 - Good to know

- Outside the scope of the Learning Objectives
- Consist of good additions to Scrum
- Covered if time allows

What happens after class?

Trainer:

1. Register participants
2. Send email to participants with link to resources (http://bit.ly/aqquCSPO)

Participants:

1. Receive mail from system@scrumalliance.org
2. Log in to scrumalliance.org
3. Create profile at Scrum Alliance
4. Generate certificate (print it)
5. Receive mail from trainer
6. Reach out to trainer for support or advice (I offer life time support of my classes)

Path to CSP

Foundational & Advanced Certifications

The progressive course model offers a staged, two-year journey that allows participants to choose an educational track that improves core Scrum knowledge, builds skill sets, and provides the necessary tools to evolve as an Agile practitioner, from Scrum foundations to advanced practice.

See also: https://www.scrumalliance.org/get-certified/certifications2017

PART 1
NEED TO KNOW

Scrum Overview

Scrum

Scrum

an **agile** framework that can be used to manage and control complex software and product development using iterative, incremental practices, which can significantly **increase productivity** and reduce time to benefits while facilitating adaptive, empirical systems development.

What is Scrum?

- Empirical framework for managing and controlling the development and deployment of complex products
- Empiricism is dependent on frequent inspection and adaptation to reach goal
- Inspection is dependent on transparency

Inspect and Adapt

- Three pillars uphold every implementation of empirical process control:
  - Transparency
  - Inspection
  - Adaptation

Transparency

The first leg is transparency

- Transparency ensures that aspects of the process that affect the outcome must be visible to those managing the outcomes.

- Not only must these aspects be transparent, but also what is being seen must be known. That is, when someone inspecting a process believes that something is done, it must be equivalent to their definition of done.

Inspection

The second leg is inspection

- The various aspects of the process must be inspected frequently enough so that unacceptable variances in the process can be detected.

- The frequency of inspection has to take into consideration that all processes are changed by the act of inspection. A conundrum occurs when the required frequency of inspection exceeds the tolerance to inspection of the process. Fortunately, this doesn't seem to be true of software development. The other factor is the skill and diligence of the people inspecting the work results.

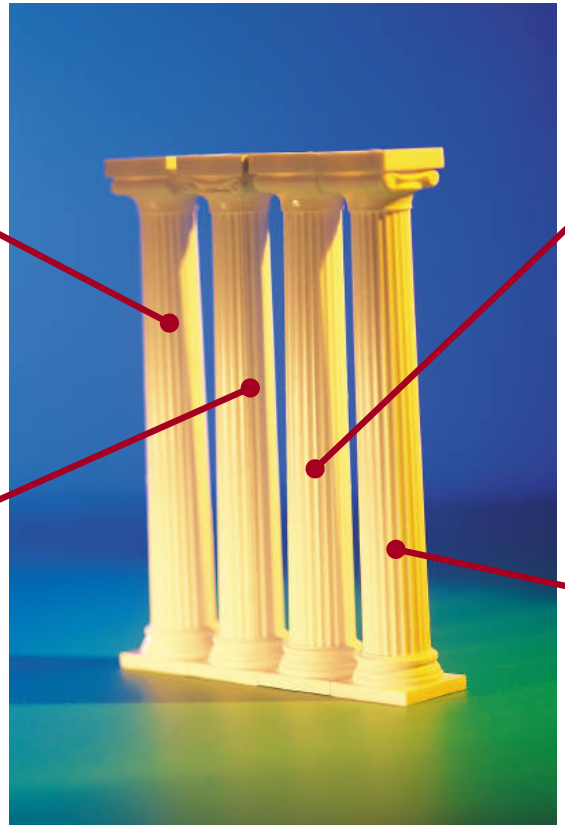
Adaptation

The third leg is adaptation

- If the inspector determines from the inspection that one or more aspects of the process are outside acceptable limits, and that the resulting product will be unacceptable, the inspector must adjust the process or the material being processed. The adjustment must be made as quickly as possible to minimize further deviation.

- There are three points for inspection and adaptation in Scrum.
  1. The Daily Scrum meeting is used to inspect progress toward the Sprint goal, and to make adaptations that optimize the value of the next work day.
  2. In addition, the Sprint Review and Planning meetings are used to inspect progress toward the Release Goal and to make adaptations that optimize the value of the next Sprint.
  3. Finally, the Sprint Retrospective is used to review the past Sprint and determine what adaptations will make the next Sprint more productive, fulfilling, and enjoyable.

Four legs of Scrum



Iterative Development

High Priority Functionality

Done Increments

Self-managing Teams that are Cross-functional

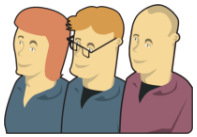
Overview of Scrum

Roles

Product Owner

ScrumMaster

Development Team



Artefacts

Product backlog
Sprint Backlog
Product Increment

Events

Sprint
Sprint Planning
Daily Scrum
Sprint Review
Retrospective

Activity:
Product Backlog Refinement

Work Culture and Scrum Values

- Work Culture
  - People collaborate,
  - enjoy their work and
  - create software that benefits its users and customers.

- Scrum Values
  - Commitment
  - Focus
  - Openness
  - Respect
  - Courage

Scrum Master – role description

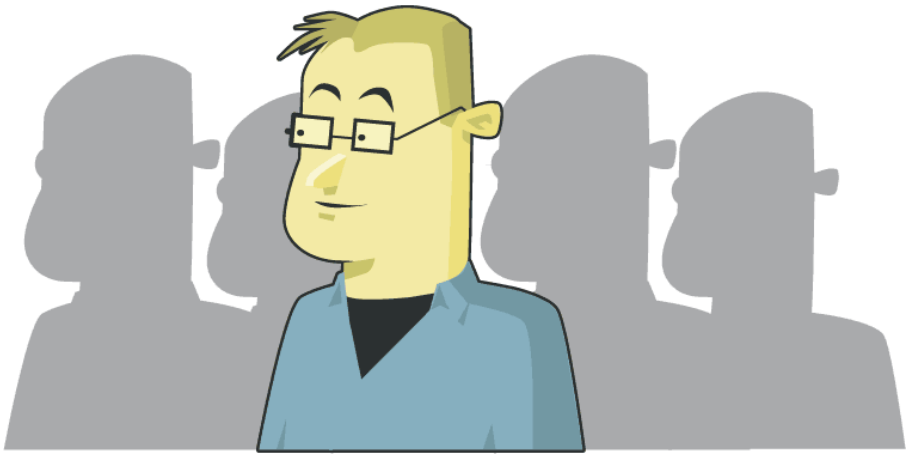
- Responsible for the success of the Scrum implementation
- Establishes Scrum practices and rules
- Shields the team
- Removes obstacles
- Is the link between management and the project

Scrum Master – responsibilities

- Teach and coach Scrum
- Facilitate meetings
- Facilitate collaboration
- Shield the team
- Remove impediments
- Improvement of Scrum implementation
- Support team, PO and organization in using Scrum

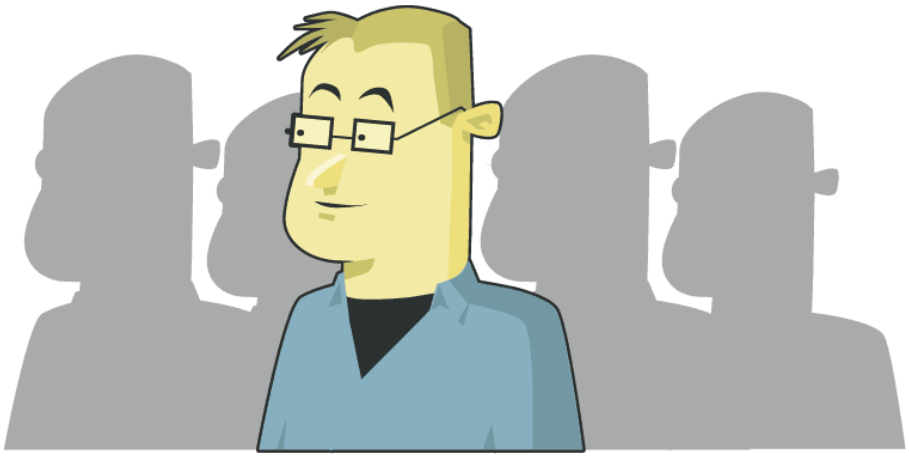
Dev Team – role description

- Three to nine (3-9) according to Scrum Guide
- Develops increments
- Self-organizing
- Cross-functional with no roles
- Responsible for committing to work
- Authority to do whatever is needed to meet commitment



Dev Team - responsibilites

- Commit to work
- Develop product increment
- Manage Sprint
- Identify impediments
- Estimate Product Backlog items
- PBL refinement
- Sprint Backlog
- Demonstrate results
- Develop ways of working



Scrum Artefacts

Required

- Product Backlog
- Sprint Backlog
- Product Increment

Other useful artefacts

- Product Vision (highly recommended by me)
- Impediments list
- List of Improvement actions
- Burndown charts

Scrum Meetings

Meetings (called events in Scrum Guide)

- Sprint Planning
- Daily Scrum
- Sprint Review
- Retrospective

Activity:

Product Backlog Refinement

The role of the Product Owner

Product Owner – role description

- Sets development schedule by prioritizing product backlog
- One person in this role ensures that only one set of requirements drives development.
- Eliminates confusion of multiple bosses, different opinions, and interference.
- Can be influenced by committees, management, customers, sales people, but is the only person that prioritizes
- To be the driving force behind product vision and goal
- Manages the releases and maximizes value of the product
- Works closely with the development team

Product Owner - Responsibilities

- Maximize product value
- Represent stakeholders
- Create & communicate Product Vision
- Product Backlog
- Release decisions (when and what)
- Available to development team
- Inspect the product increment
- Accept the result of the work

Realizing Value

One of the main responsibilites of the Product Owner is to maximize the value of the product. As an organization we want to realize value through delivering product solutions that delight customers and users.

We need to do that within the constraints of technical feasibility. The Product Owner has a big role to play to make sure that the Product Vision is possible and feasible to reach.

The Product Owner most likely cannot do this alone. Collaboration with the Development Team and other relevant stakeholders within the organization is often required.

A Good Product Owner

- Understands customer needs thoroughly
- Able to create and communicate the product vision
- Empowered to make decisions,
  is decisive and know when to say no
- Has good working conditions with
  - the stakeholders
  - the team/s
- Understands value creation

A Great Product Owner:

- understands the desires/problems the solution (product/service) should serve - and knows about techniques to get there
- can formulate the solution in a compelling product vision
- can gather an inspired team and create a shared understanding of the product vision
- knows about techniques on how to identify solutions e.g. Design Thinking
- knows that every product idea needs to be validated, thus is familiar with the concept of MVPs and running experiments
- defines clear outcomes/goals as a measure of success
- breaks down the product vision into digestible Product Backlog Items - or makes sure this gets done
- prioritizes the team's work through prioritizing the Product Backlog
- regularly communicates with his team and other stakeholders to reevaluate the state of the product
- creates ownership for the product in others... this means he educates his team about the product, customers, and the business aspects of it.

Courtesy Sohrab Salimi

Product Owner anti-patterns

Avoid the following Product Owner anti-patterns:

- Product Owner is an order taker without decision power
- Product Owner wants it all, not dealing with details
- Product Owner is not present or available at all

Organizational Context

As Product Owner you may need to act differently in dífferent organizational contexts. Thus, pay attention to your organizational context.

For example, you as Product Owner:

- have full ownership of the product, including customer, what problem the product is solving and how the solution is designed
- are managing the production and delivery of an idea or solution from someone else
- deliver to other teams in the organization

PO and Stakeholders

A big part of your responsibilities as a Product owner is representing stakeholders.

In order to provide transparency towards stakeholders the Product Owner may use techniques like

- Road map – to communicate direction of the product
- Sprint reviews – to receive feedback
- Release burn-up chart tp communicate progress

When engaging with stakeholders you need techniques to gather information and insights. Techniques demonstrated during the workshop were: roman evaluation, dot voting, fist of five, and affinity grouping.

PO and the Development Team

Product Owner is a collaborative role.

Collaboration with the Development Team involves:

- Agreeing upon definition of Done
- Creating the Product Backlog
- Refining the Product Backlog
- Ordering the Product Backlog

Working with multiple teams

A Product Owner may need to work with more than one team. When doing so, dependencies between teams need to be visualized, managed and reduced.

A Product Owner may use different techniques to obtain the above:

1. Coordinate with other Product Owners
2. Redefine Product Backlog items to remove dependencies
3. Ensure Product Backlogs are visibly shared between Product Owners and Development Teams

Purpose and Strategy

Words, terms & definitions

Purpose

Vision

Mission

Tactics

Strategy

In every product development organization it's essential to agree upon how to use the terms above. So far, there is not one "right" definition for these words. Each organization needs to come up with there own to reach clarity.

Why a Product Vision?

- The product backlog represents the overall product capabilities
  1. Assumed to change and emerge
- The containing stories represent individual capabilities – "chunks" of requirements
- Behind every backlog is an overall product vision to
  - Guide the teams
  - Help teams remain focused on the critical aspects of the product
  - Align stakeholders and customers

A Good Product Vision

- **Shared**
  - Everyone must buy into it.
- **Stable**
  - It should stay constant during the course of the project or release.
- **Clear**
  - It should be easy to understand the vision.
- **Broad and engaging**
  - The vision should provide guidance for the team while leaving room for creativity and trade-offs.
- **Concise**
  - It is often captured on one or two flip chart pages or on a wiki web page

Product Vision

Short Version

From *Crossing the Chasm* by Geoffrey Moore

1. For <*target customer*>
2. Who <*statement of need or opportunity*>
3. The <*product*> is a <*product category*>
4. That <*key benefit or compelling reason to buy/use*>
5. Unlike <*primary competitors/competition*>
6. Our product <*statement of primary differentiators*>.

How to shape a Product Vision

Techniques to try:

1. Vision Box
2. Elevator Pitch
3. Press Release
4. Magazine Review
5. One pager

Design-the-Box exercise

Assumption: the product will be sold in a shrink-wrapped box

Task: design the product box front and back

1. product name
2. a graphic
3. three to four key bullet points on the front to "sell" the product
4. a detailed feature description on the back
5. operating requirements

Following Steps

After a Design-the-Box workshop the following may be done:

1. Constructing your own version of Moore's product vision statement

2. Create a complete one- to three-page product vision document that might include:
   - the mission statement,
   - a project profile overview (scope, schedule, cost, defects) with priorities,
   - pictures of the "boxes”,
   - target customers and each of their needs,
   - customer satisfaction measures,
   - key technology and operational requirements,
   - critical product constraints (performance, ease of use, volumes),
   - and key financial indicators

Product Vision – examples 1

- "to organize the world's information and make it universally accessible and useful”
- ”to build a place where people can come to find and discover anything they might want to buy”
- ”For people worldwide who are interested in Scrum, the new Scrum Alliance website will be their trusted source of Scrum knowledge. It will be featureand content rich and will be their first step on the Internet for learning more about Scrum or to collaborate on Scrum topic of interest.”

Product Vision – examples 2

” For post-production film engineers

Who are dissatisfied with the limitations of traditional film editors

Our workstation is a digital film editor

That lets you modify film images any way you choose.

Unlike workstations from Sun, HP, or IBM,

We have assembled all the interfaces needed for post-production film editing.”

- ” The iPod will be a portable digital music player that will hold 5000 songs. It will have a battery life measured in days, not hours. You will navigate the thousands of songs with a single finger. You will sync all your music from your computer to the iPod in minutes automatically, so you can have all your music in your pocket.”

Communicating the Product Idea

The purpose of havinga product idea (aka product vision) is to clearly communicate

- what **problem** is **being solved** by the product
- who is more **affected** by the problem
- how the Development Team's effort will **improve** the situation
- how the effectiveness of the solution is **evaluated**

Roadmaps and Release Planning

Roadmaps and Release Planning are optional in Scrum. In certain circumstances both Roapmaps and Release Plans can add value to the Scrum Framework.

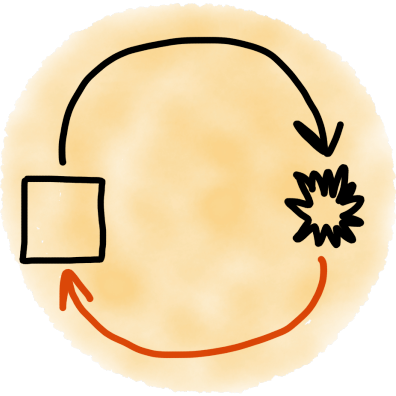
In Scrum, the Product Owner is asked to put together a Product Backlog and keep it relevant and up to date. The Development Team is asked to put together a Sprint Backlog for each Sprint.

It is possible for the Product Owner to create a Roadmap for the product and a Release Plan for each release. These would be additional artifacts to be used if they add value to the set of artifacts prescribed by Scrum.

If the Product Owner finds it valuable to use a Roadmap or a Release Plan, those can often be derived from the Product Backlog.

Product Discovery

What is Product Discovery?



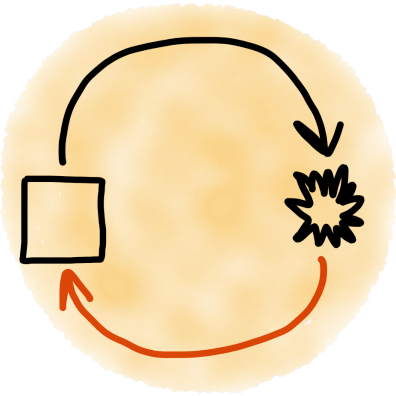
Product discovery is about getting early feedback.

The term was coined by Marty Cagan. He writes:

"I prefer to think of this phase as "product discovery" more than "requirements and design." I think this nomenclature emphasizes two all-important points:

1. First, you need to discover whether there are real users out there that want this product. In other words, you need to identify your market and validate the opportunity with your customers.
2. Second, you need to discover a product solution to this problem that is usable, useful, and feasible. In other words, you need to design your product and validate it with your customers and your engineering team.

Driving Product Discovery?

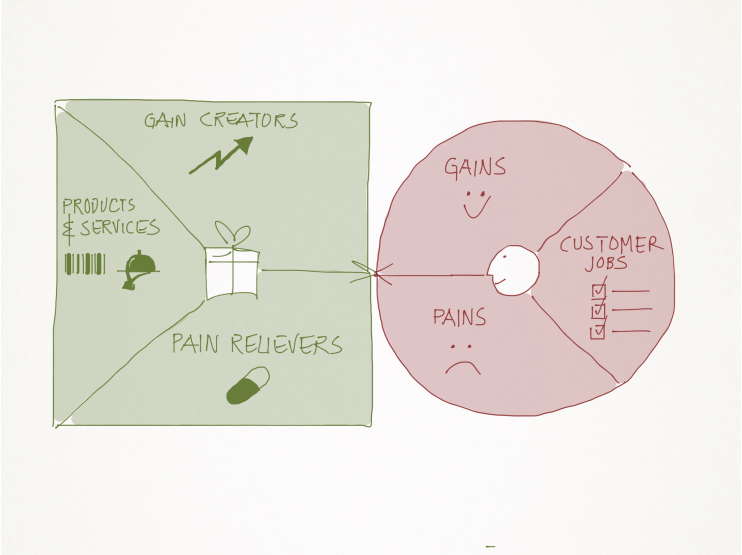


In order to drive Product Discovery, the following three questions are good to ask and answer.

1. What problem are we trying to solve?
2. Who are we solving the problem for?
3. How will we know if you have succeeded?

Value Proposition Canvas
Business Model Canvas
Lean Canvas
Product Scorecard
Product Vision Board
Empathy Map
Stakeholder Map
Impact Mapping
Innovation Games
Lean Startup
Story Mapping

Value Proposition Canvas



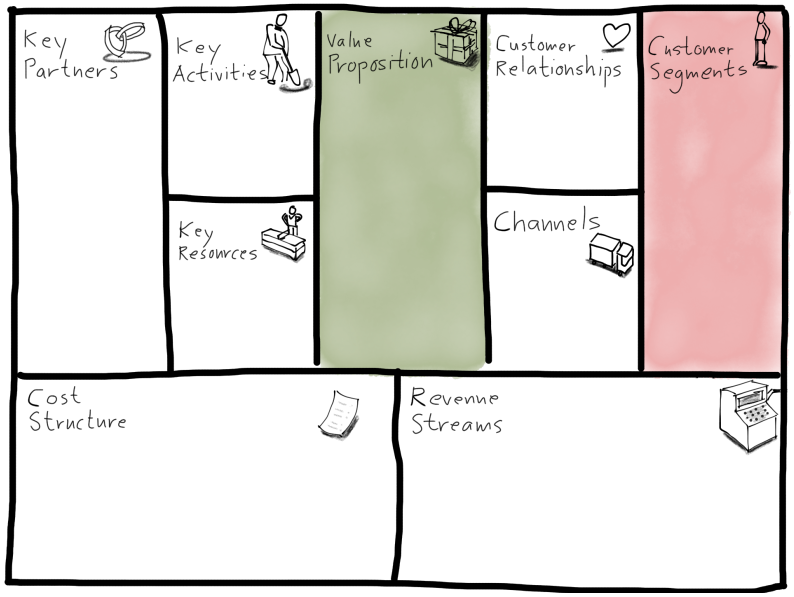
A Value Proposition Canvas is a tool to visualize two basic things:

1. The Customer Segment that you are focusing on creating value for
2. The Value Proposition that you believe will make your customers wanting to buy your product or use your service

You should aim to show the fit between what you offer and what customers want.

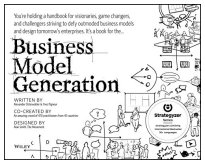
Value Proposition Design: How to Create Products and Services Customers Want (Strategyzer)
by Alexander Osterwalder et al.
Link: http://a.co/4eZIT5C

Business Model Canvas

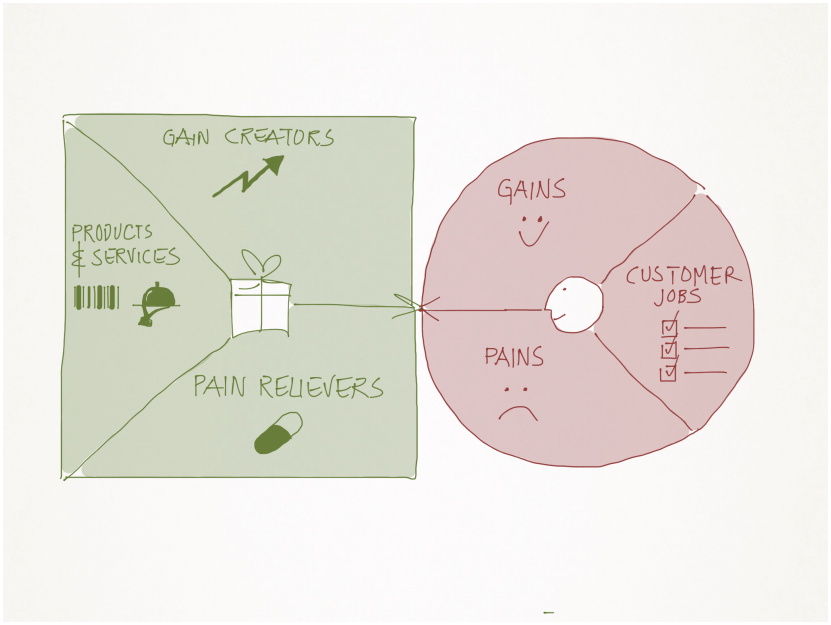


The Business Model Canvas is a strategic management and Entrepreneural tool.

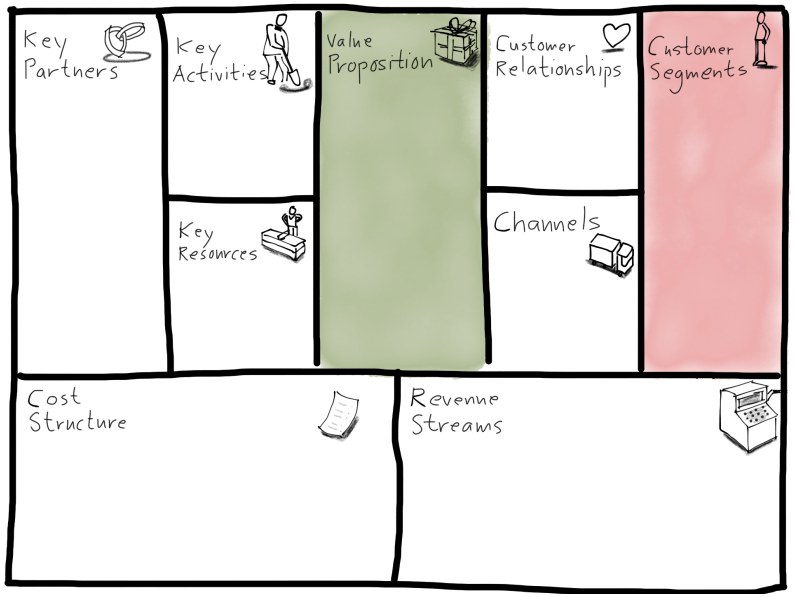
It allows you to describe, challenge, invent, and pivot your business model.



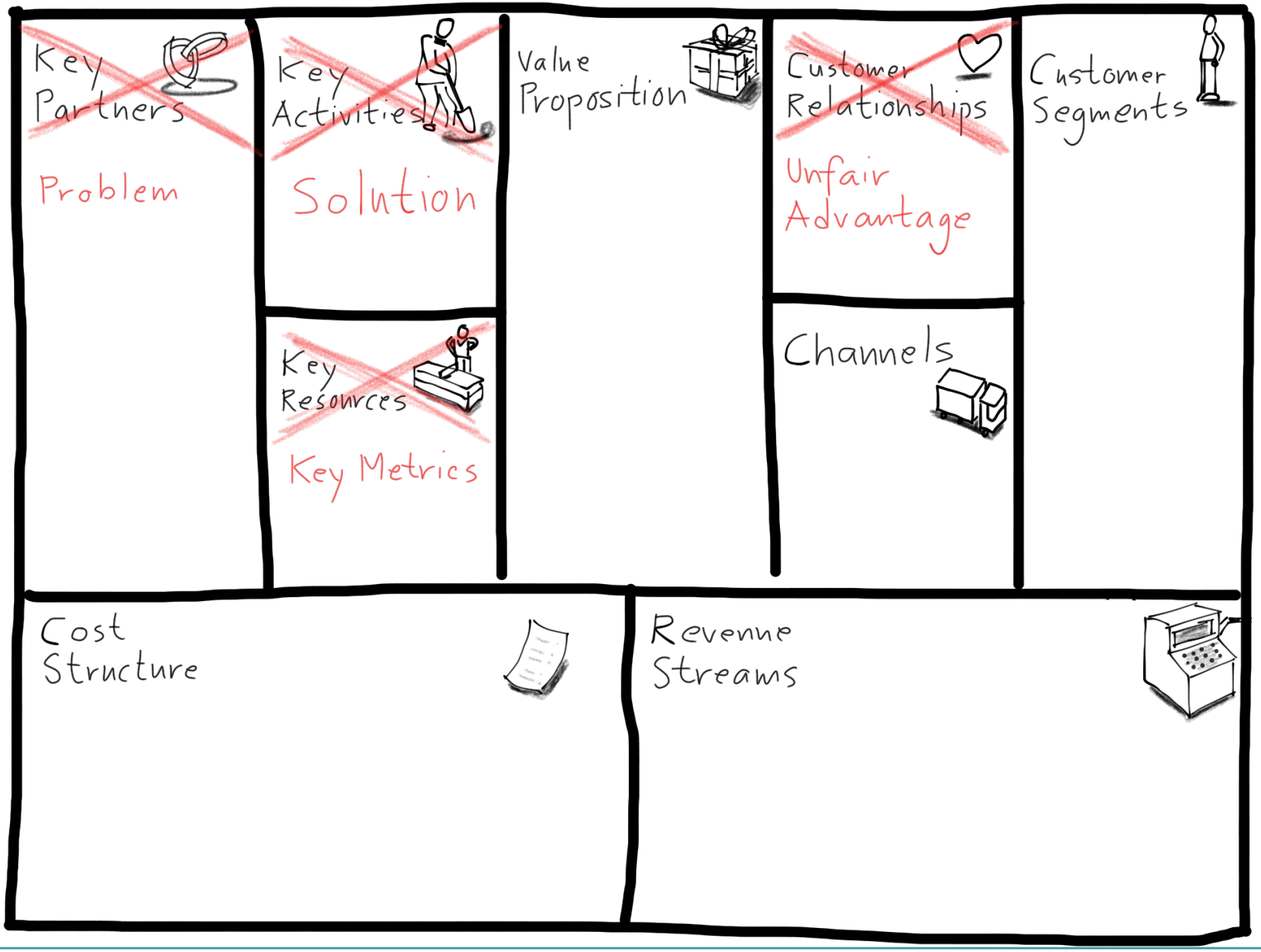
Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers
by Alexander Osterwalder, Yves Pigneur



The Value Proposition Canvas fits well into the Business Model Canvas. Both invented by Alex Osterwalder and described in the books ***Business Model Generation*** and ***Value Proposition Design***.
https://strategyzer.com/canvas



Lean Canvas created by Ash Maurya.
http://practicetrumpstheory.com/why-lean-canvas/



ROMAN'S PRODUCT SCORECARD

pichler consulting

BUSINESS GOALS

The desired business benefits the product should create; prioritised and measurable.

FINANCIAL

The relevant financial indicators, for instance, revenue, cost, profit.

CUSTOMER

The relevant customer indicators, such as market share, adoption rate, referrals.

PRODUCT AND PROCESS

Product and process indicators including product health and schedule variances.

PEOPLE

People indicators, like team motivation and stakeholder engagement.

THE PRODUCT VISION BOARD

pichler consulting

VISION

What is your motivation 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?

THE PRODUCT VISION BOARD EXTENDED

pichler consulting

VISION

What is your motivation 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

Which problem does the product solve?

What 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?

COMPETITORS

Who are your main competitors?

What are their strengths and weaknesses?

REVENUE STREAMS

How can you monetise your product and generate revenues?

COST FACTORS

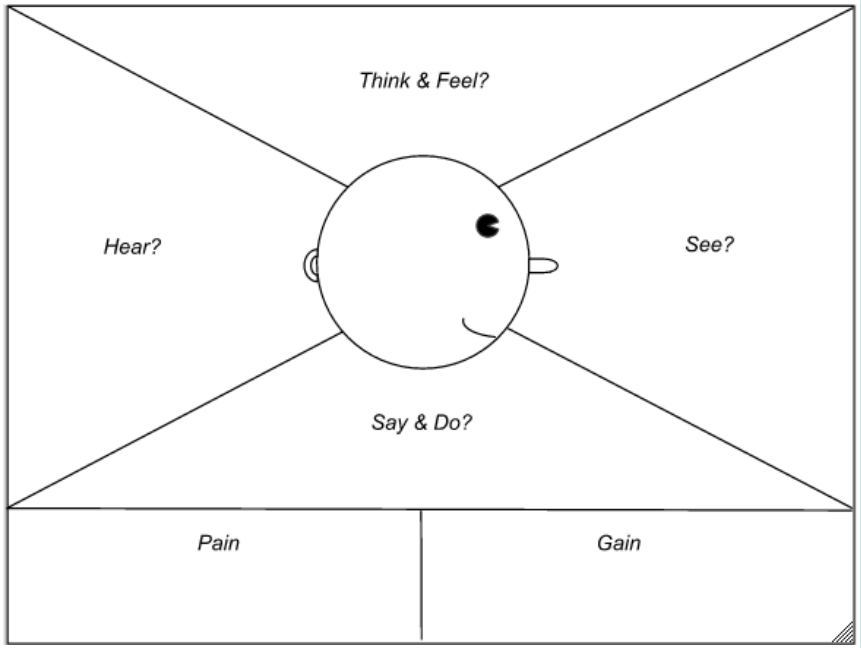
What are the main cost factors to develop, market, sell, and service the product?

CHANNELS

How will you market and sell your product?

Do the channels exist today?

Empathy Map



An Empathy Map can help you gain a deeper level of understanding of a stakeholder in your business ecosystem, which may be a client, prospect, partner, etc., within a given context, such as a buying decision or an experience using a product or service.

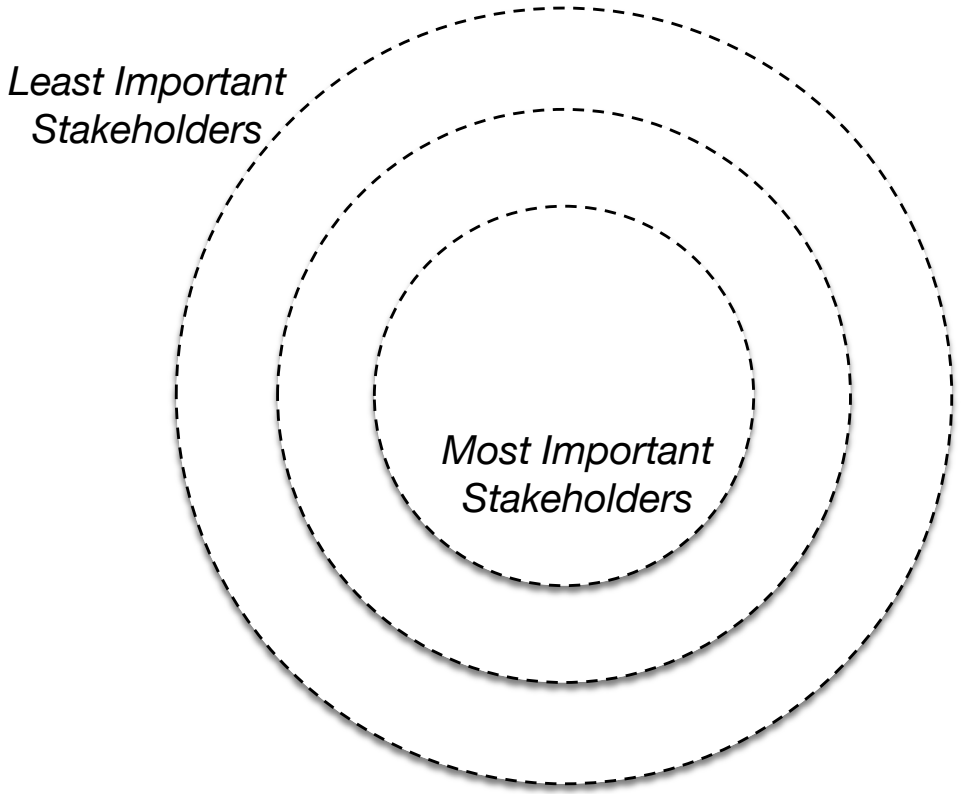
The exercise can be as simple or complex as you want to make it. You should be able to make a rough empathy map in about 20 minutes, provided you have a decent understanding of the person and context you want to map.

Even if you don't understand the stakeholder very well, the empathy-mapping exercise can help you identify gaps in your understanding and help you gain a deeper understanding of the things you don't yet know.

Download a copy of this canvas at http://gamestorming.com/empathy-map/

Stakeholder Map

A Stakeholder Map is a tool to visualize and understand how your stakeholders are connected to each other, analyze how they influence your product or service, and use the insights to optimize your ecosystem.



Impact Mapping

Why? – are we doing this? **GOAL**

Who? – can produce the desired effect? **ACTORS**
– can obstruct it?
– are our customers?
– will impacted?

How? – should our actors' behaviour change? **IMPACT**
– can they help us achieve the goal?
– can obstruct or prevent us?

What? – can we do, as an organization or a delivery team, to support the required impacts? **DELIVERABLES**

Innovation Games

Innovation Games

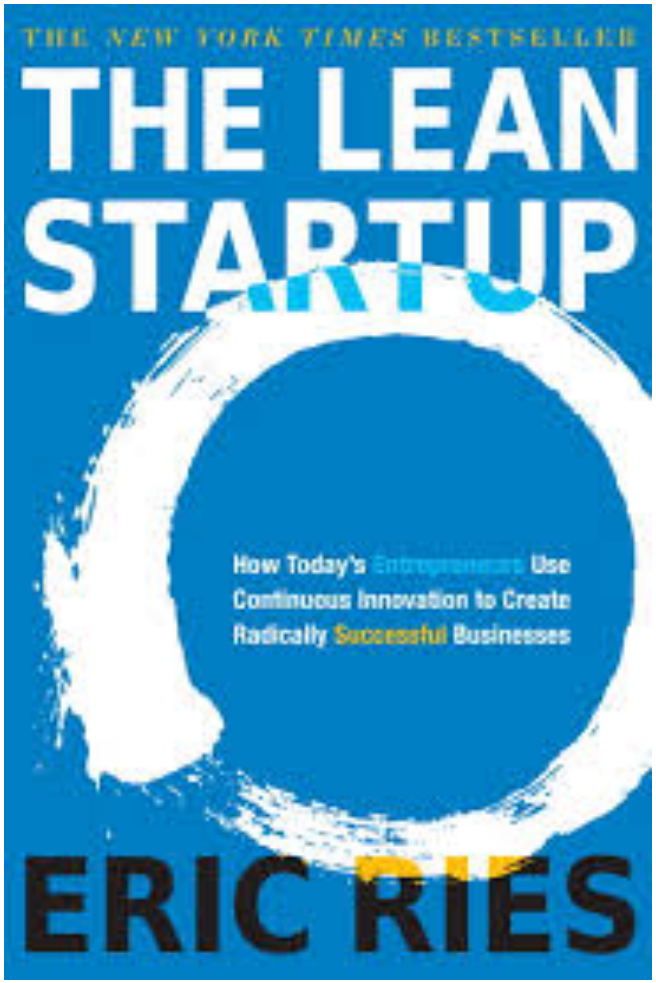
In the book Innovation Games, Luke Hohmann, presents different games and exercises that are beneficial for product owners.

The ones that I find most beneficial are listed to the right

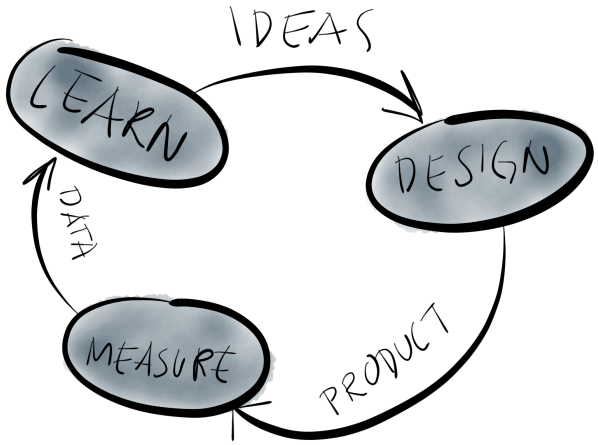
- Product Box
  (aka Vision Box)
- 20/20 Vision
- Prune the Product Tree
- Buy a Feature
- Speed Boat

https://www.innovationgames.com/the-innovation-games/

Lean Startup



Lean Startup



- Eliminate Uncertainty
- Work Smarter not Harder
- Develop an MVP
- Innovation
- Validated Learning

The purpose behind Lean Startup is to learn without spending too much time, effort and money.

Starting with an idea you build or design a product, measure how it's received and learn if your product idea is viable or not.

The concept of MVP is central. Different types of MVPs are used.

A concierge MVP is a method of manually guiding your user through your solution.

If using the "Wizard of Oz" MVP the customer doesn't see all the manual work. Your website or app look and "feel" like real products, but you carry out all the product functions manually.

Story Mapping

Story mapping was coined by Jeff Patton.

The purpose of using Strory Maps are twofold:

1. To add a dimension to the Product Backlog. A Product backlog in Scrum is often ordered based upon Value. It hepls showing the Return of Investment dimension. When creating a Story map we are also interested in the usage dimension. What is the first thing a user would do? What is the next thing? What is the last thing?
2. To create a shared understanding of what product we are supposed to build. Traditionally the understanding was believed to be conveyed by the requirements. That is not always enough. In Scrum we say that we should write less, talk more, and write as much as necessary.

Testing Product Assumptions

Assumption

An assumption is any statement that is believed to be true.

Hypothesis

A hypothesis is an argument put forward to explain a phenomenon or sets of phenomena

Testing Product Assumptions in Scrum

In Scrum we are using short feedback loops to test product assumptions.

Every Sprint we expect the Development Team to present a working potentially shippable product increment.

Thus, the Sprint Review is one instance where we test product assumptions in Scrum

Working with the Product Backlog

Outcome versus Output

When working with the Product Backlog two sentences from the Scrum Guide are important to remember.

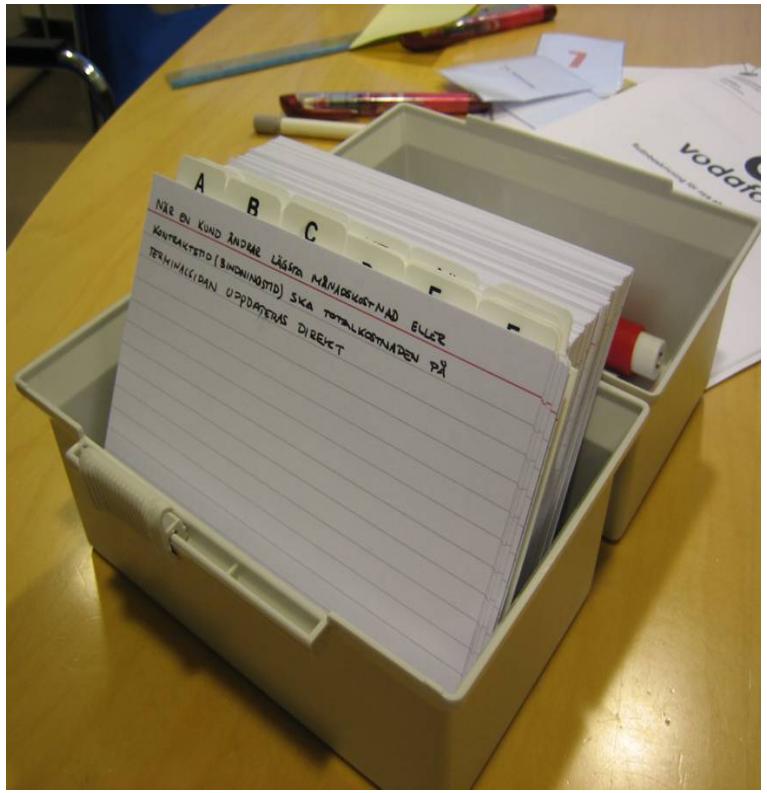
1. The Product Owner is responsible for maximizing the value of the product resulting from work of the Development Team.
2. Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next (as one element of the Sprint Review)

These two sentences focus on **outcome** rather than output. **Outcome** is the value provided by the product or product increment being developed and offered.

Outcome is what the Product Owner is set to maximize. The tool to use is the Product backlog. By ordering the items one Product backlog the Product Owner can facilitate maximizing the **outcome**.

Product Backlog

- The requirements
- A list of all desired work to get a desired product
- Ideally expressed so that each item has a value to users or customers of the product
- Owned and prioritized by the Product Owner
- Reprioritized at the start of every sprint

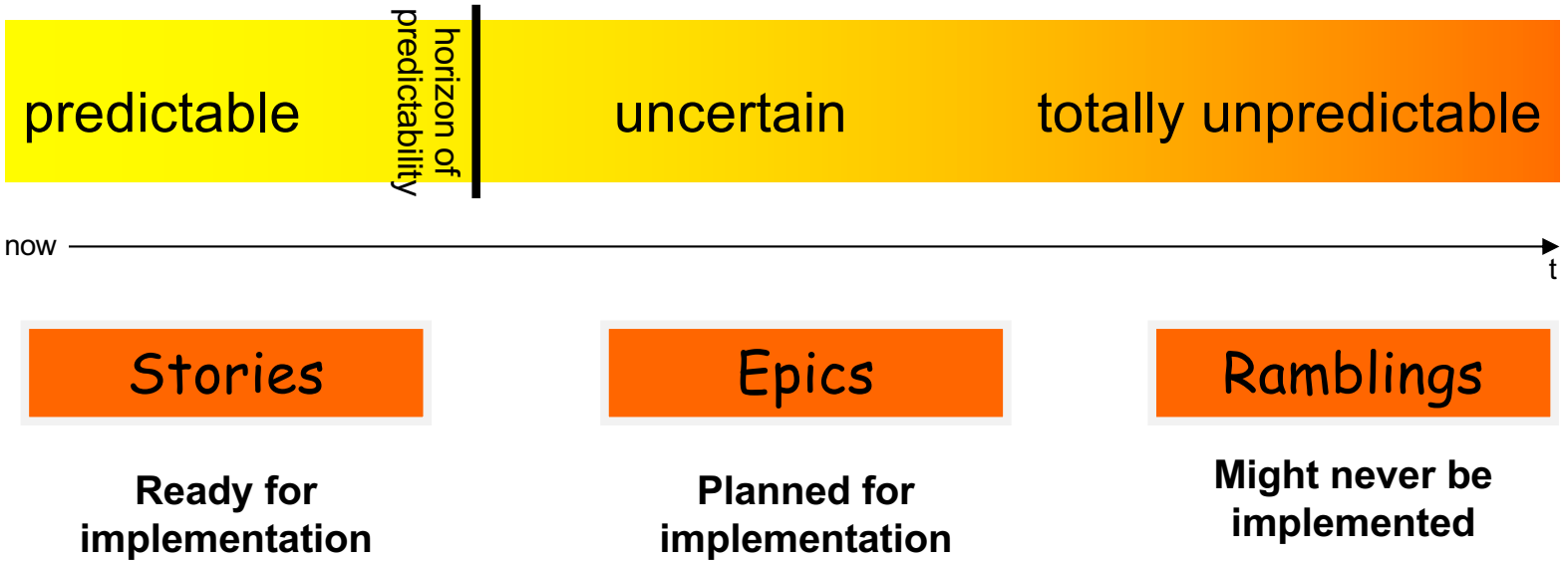


A sample Product Backlog

Backlog Item	Estimate
Allow a visitor to reserve tickets	3
As a visitor, I want to cancel a reservation	5
As a visitor I want to change seats	3
As a stadium employee, I can run reports	8
Improve error handling	8
…	20
…	40

How much detail?

Just enough…



Detailed to the horizon of predictability

Defining Value

- **What is Value?**

In management, business value is an informal term that includes all forms of value that determine the health and well-being of the firm in the long-run.

Business value expands concept of value of the firm beyond economic to include other forms of value such as employee value, customer value, supplier value, channel partner value, alliance partner value, managerial value, and societal value. Many of these forms of value are not directly measured in monetary terms.

Reference: Wikipedia

- What is NOT Value?

Everything which is not considered to be business value. Like size, amount of effort, how much, etc.

- Why do we need to know Value?

We want the Product Owner to maximize value of the product.

Different types of Business Value

- **Sales**
  - New sales
  - Incremental (existing)
  - Retained
- **Operational efficiency** (savings)
- **Customer satisfaction**

PRAISED – other view on Business Value

Productivity gains

Reduced cost

Avoided cost

Increased revenue

Service level improvements

Enhanced quality

Differentiation in the marketplace

Businesses are quite simply looking for one thing, ***RESULTS***! These results generally involve some monetary quantification i.e. they can be measured in real dollars. In general the results the businesses are looking for can be broadly categorised as ***PRAISED*** items as described to the left

If a software development project can deliver on one or more of the PRAISED items then it will have delivered real business value back to the business as a whole

Measuring Value

Value and Business Value can be measured. All of the examples on the previous pages can be measured separately. Some of them more objectively.

The challenge will come when we want to compare different types of Business Value.

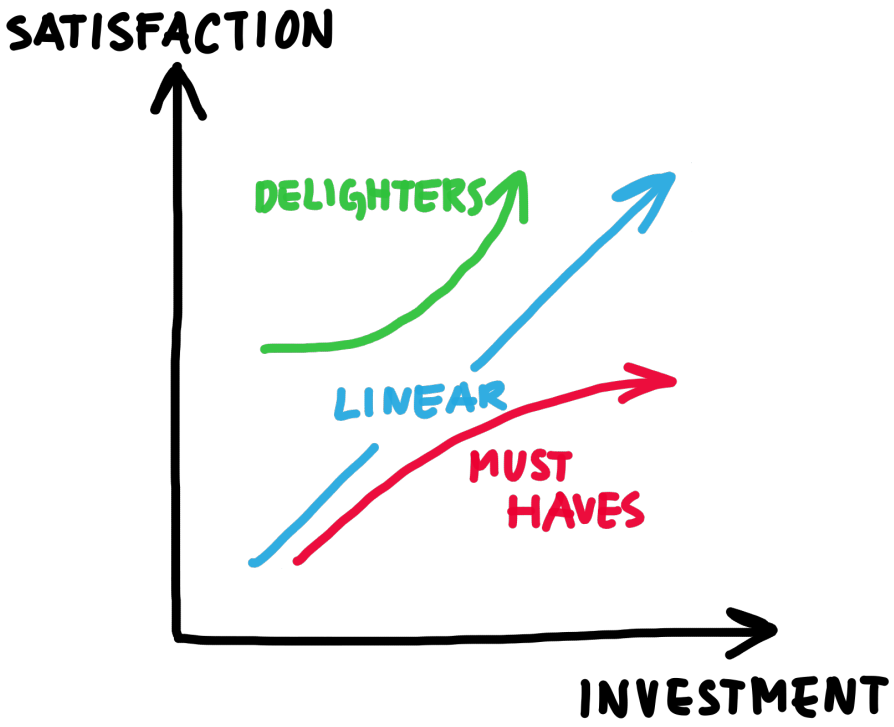
Value from different perspectives

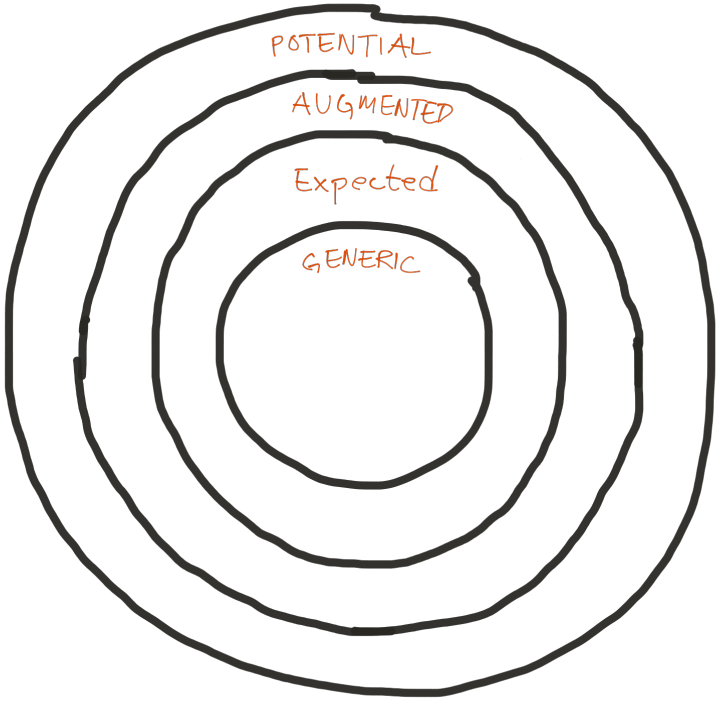
1. User
2. Customers
3. Stakeholders
   a. Internal
   b. External
4. Dev Team

Value Methods

On the following pages some methods are mentioned and some described. Others are just listed below

- Financial analysis
- Kano analysis - mentioned
- Whole Product Model - mentioned
- **Feature screening** - described
- **Feature scoring** - described
- **Relative weighting** - described
- Analytical hierarchical process
- Evaluation of business potential
- Value Scorecard
- WBS based Business Value


SATISFACTION
DELIGHTERS
LINEAR
MUST HAVES
INVESTMENT


POTENTIAL
AUGMENTED
Expected
GENERIC

Feature screening

Identify 3-7 (approximately) selection criteria for what is important in the next release

Select a baseline feature

1. Likely to be included in the next release
2. Understood by most team members

Asess each candidate feature relative to the baseline feature

Feature screening - Example

Selection Criteria	Features						
	Feature 1	Feature 2	Feature 3	Baseline Feature	Feature 4	Feature 5	Feature 6
Importance to existing customers	+	+	+	0	0	+	0
Competitiveness with Comp X	-	0	-	0	0	0	-
Makes maintenance easier	+	+	0	0	+	-	0
Generates revenues in Q3	0	+	0	0	+	0	0
Net Score	1	3	0	0	2	0	-1
Rank	3	1	4	4	2	4	7
Continue?	Y	Y	N	N	Y	N	N

\+ = better than
0 = same as
\- = worse than

Feature scoring

Like feature screening but selection criteria are weighted

Need to select a baseline feature for each criteria

1. Avoids compression of a category

Each feature is assessed against the baseline for each selection criteria

Much worse than reference	**1**
Worse than reference	**2**
Same as reference	**3**
Better than reference	**4**
Much better than reference	**5**

Feature scoring - Example

Selection Criteria	Weight	Feature 1		Feature 2		Feature 3	
		Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Importance to existing cust.	20	1	0.20	3	0.60	4	0.80
Competitive. W. Comp X	10	3	0.30	2	0.20	3	0.30
Makes maint. easier	20	4	0.80	3	0.60	4	0.80
Generates Q3 revenue	50	2	1.00	5	2.50	3	1.50
Net Score			2.30		3.90		3.40
Rank			3		1		2
Continue?			No		Yes		Yes

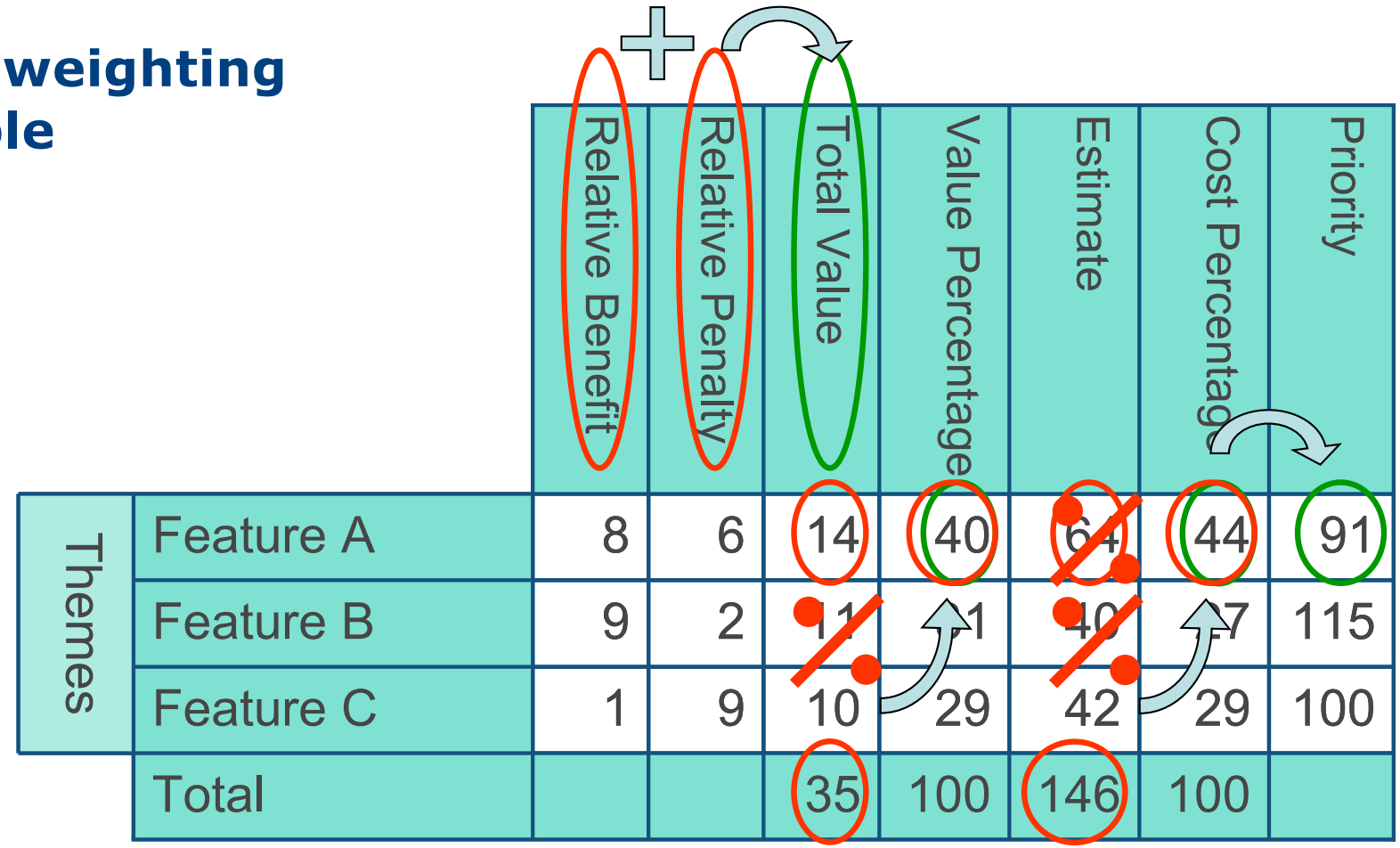
Relative weighting

Prioritizing based on Relative value and Relative cost

How to

1. Assess the impact of implementing a req from 1-9
2. Assess impact of NOT implementing it from 1-9
3. Calculate the value of each req relative to the entire product backlog
   - This gives you the relative value of that requirement
4. Estimate the cost of each requirement
5. Calculate the cost of each req relative to the entire product backlog
   - This gives the relative cost of that requirement
6. Priority is given by (Relative Value / Relative Cost)

Relative weighting - Example



Themes	Relative Benefit	Relative Penalty	Total Value	Value Percentage	Estimate	Cost Percentage	Priority
Feature A	8	6	14	40	64	44	91
Feature B	9	2	11	31	40	27	115
Feature C	1	9	10	29	42	29	100
Total			35	100	146	100	

Total Value = Relative Benefit + Relative Penalty
Value Percent = Total Value / Σ (Total Value)
Cost Percent = Estimate / Σ (Estimate)
Priority = Value % / Cost %

Ordering
Product Backlog Items

One of the main responisibilities of the Product Owner is to maximize the value of the product.

The Product Owner can use the ordering of the items of the Product Backlog to communicate what is most importyant to do firts and what should come after that.

Ordering or Prioritizing?

Factors to consider:

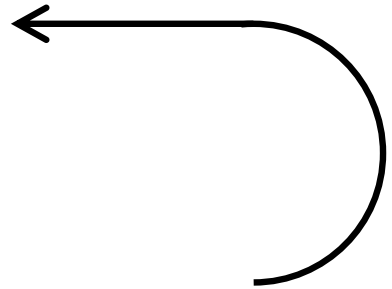
1. The **business value** of having the features.
2. The **cost of developing** (and perhaps supporting) the new features.
3. The **amount and significance of learning and new knowledge** created by developing the features.
4. The **amount of risk removed** by developing the features.
5. **Dependencies**
6. **Cost of Delay**

Progressive refinement

- Product Backlog items that are further into the future can be larger
- Each PBI should be described in just enough detail that the team can complete it in one sprint
  1. May attach things like:
     - User interface design
     - Mathematical algorithms
     - Tests
- Strive to describe each item as briefly as possible

One way of doing it

1. Identify/Define
2. Value
3. Estimate
4. Prioritize
5. Break down



If an item is too large to bring into next Sprint it needs to be broken down and then the parts need to be valued, estimated and prioritized

Input to Product Backlog

Requirements

Features

User Stories

Use Cases

Possible sources are:

- Users
- Customers
- Development Team
- Stakeholders
  - Authorities
  - Regulatory authorities
  - Standardization bodies

A Product Backlog should be DEEP

Detailed appropriately

Estimated

Emergent

Prioritized

INVEST in your product backlog

- **I**ndependent
- **N**egotiable
- **V**aluable
- **E**stimatable
- **S**ized appropriately => Small
- **T**estable

Creating and Refining Product Backlog Items

In Scrum the creation and refinement of product backlog items is done differently compared to traditional devlopment.

We move from a traditional up front approch to a just in time approach.

Traditional Development

- Prior to Scrum and Agile, it was considered prudent to understand what was wanted and how it was going to be delivered at the very start of the project

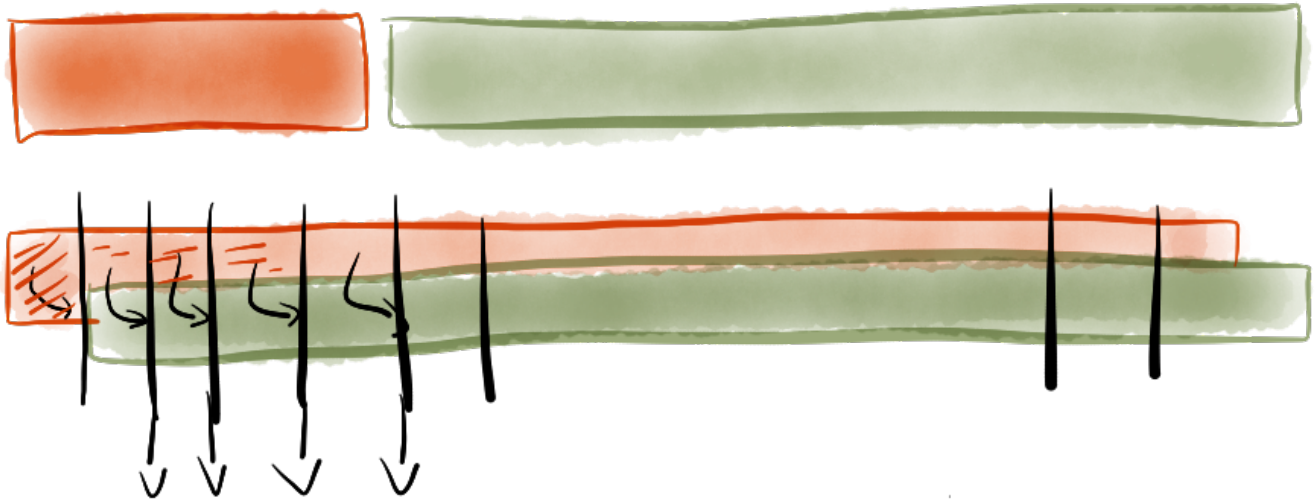
- All future work was dependent on the work to understand

What are the risks with this approach?

- **Possible answer**:
  Trying to be complete rather than thinking of value

Agile Development

- The Art of the Possible
- In Scrum we start building the product before we have detailed all requirements



- We do it Just In Time
- Good enough

Summary

Product Owner Collaboration

The Product Owner is a collaborative role. PO collaboration can be described in terms of What, Who and When.

What

- Product Vision
- Requirements/Stories/ Acceptance Criteria
- Sprint Goal
- Product Backlog
  - Creation
  - Refinement
- Definition of Done
- Product Increment
- (Release Plan)

With whom

- Stakeholders
- Development Team
- ScrumMaster

When

- Before Sprint
  - Product Vision
    - Creation
    - Communication
  - Estimation
  - Product Backlog
  - (Release Planning)
- During Sprint
  - Support to Development Team
  - Giving feedback to Development Team
  - Answering questions
- After Sprint
  - Feedback from stakeholders (mainly during Sprint Review)
  - Reporting to Stakeholders

Product Owner Toolbox

A Product Owner benefits from understanding what tools, practices, and techniques that can be useful. I call them all Tools and I believe it’s good for the PO to be aware of what tools to have in the PO Toolbox.

It’s also recommended for a PO to constantly try out new tools, try to improve existing tools, and experiment with new tools.

Advice to a new Product Owner

For a list of advice to a new PO I refer to my blog:
http://aqqurite.se/agile/advice-to-a-new-product-owner

The advice listed (with elaborations) are:

- Use time to understand the role
- Understand other roles and the Scrum framework
- Make sure that you spend enough time to perform the role
- Focus on communication
- Be sure to know and understand product and business needs
- Get to know the Product Stakeholders
- Be very systematic in your role
- Dare to demand – stand up for your role
- Continuously refine and update Product Backlog together with Stakeholders and Team

Scrum Resources

Aqqurite (http://aqqurite.se/material-from-certified-scrum-product-owner-training)

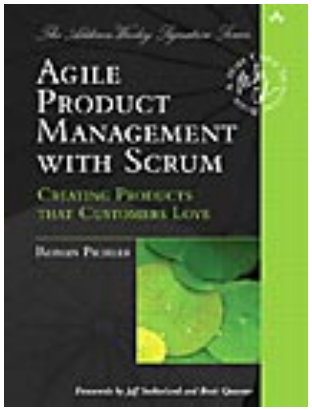
The Scrum Alliance (www.scrumalliance.org)

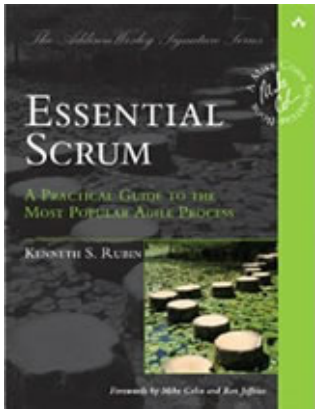
All Things Product Owner (www.allthingsproductowner.com/)

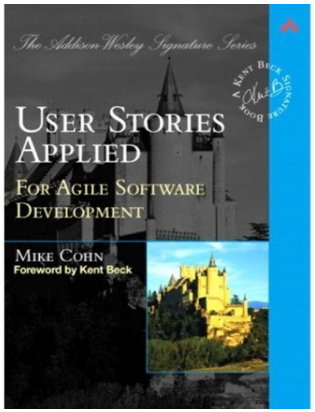
Mike Cohn (www.mountaingoatsoftware.com)

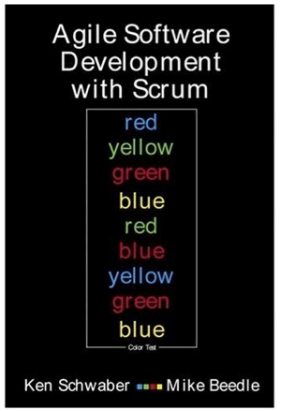
Agile Alliance (www.agilealliance.org)

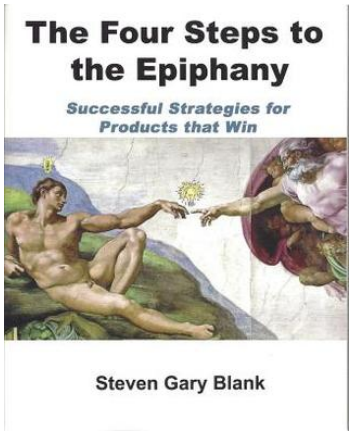
Literature

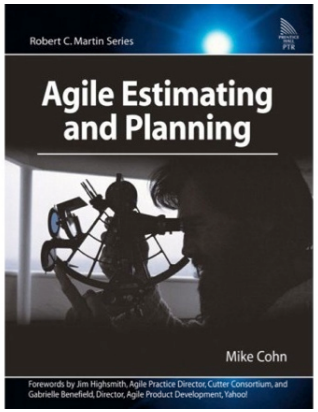

Agile Product Management with Scrum
Creating Products that Customers Love

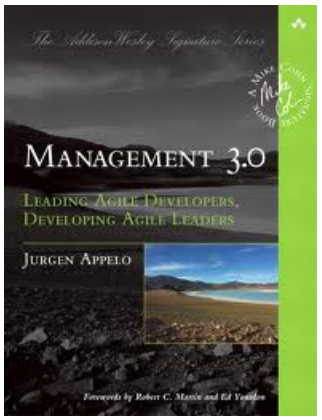

Essential Scrum
A Practical Guide to the Most Popular Agile Process
Kenneth S. Rubin

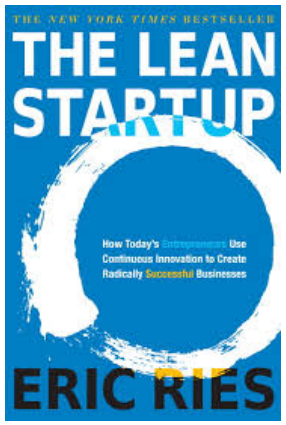

The Addison Wesley Signature Series
User Stories Applied
For Agile Software Development
Mike Cohn
Foreword by Kent Beck

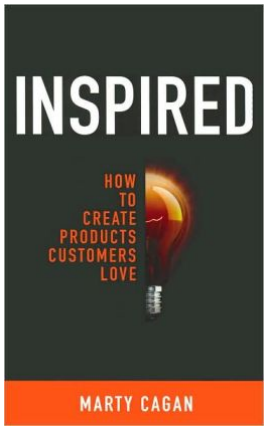

Agile Software Development with Scrum
red
yellow
green
blue
red
blue
yellow
green
blue
Ken Schwaber
Mike Beedle

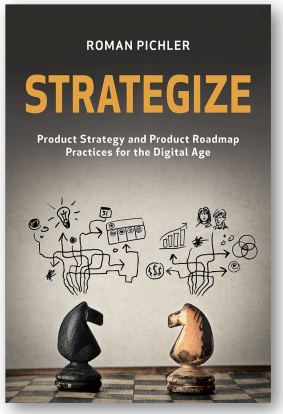

The Four Steps to the Epiphany
Successful Strategies for Products that Win
Steven Gary Blank


Robert C. Martin Series
Agile Estimating and Planning
Mike Cohn


Management 3.0
Leading Agile Developers, Developing Agile Leaders
Jurgen Appelo


The New York Times Bestseller
The Lean Startup
How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses
Eric Ries


INSPIRED
How to Create Products Customers Love
Marty Cagan


Roman Pichler
Strategize
Product Strategy and Product Roadmap Practices for the Digital Age

PART 2
GOOD TO KNOW

Agile principles for spec's and doc's

We don't want write-only documentation

We don't want to throw documentation over-the-wall

We want "just enough"

You are the only one who knows what "just enough" means for your product in your organization!

Emergence

- It is impossible to know all requirements in advance
- ”Thinking harder” and ”thinking longer” can uncover some requirements, but…

Every project has some emergent requirements

- …Emergent requirements are those users cannot identify in advance

How do we handle emergence?

Talk more, write less

1. write some if you need to

Show software to users

Acknowledge that requirements emerge

1. and all that this implies

Progressively refine our understanding of the product

1. And express this progressive refinement in the product backlog

User stories on the backlog

Card

1. traditionally written on note cards
2. may be annotated

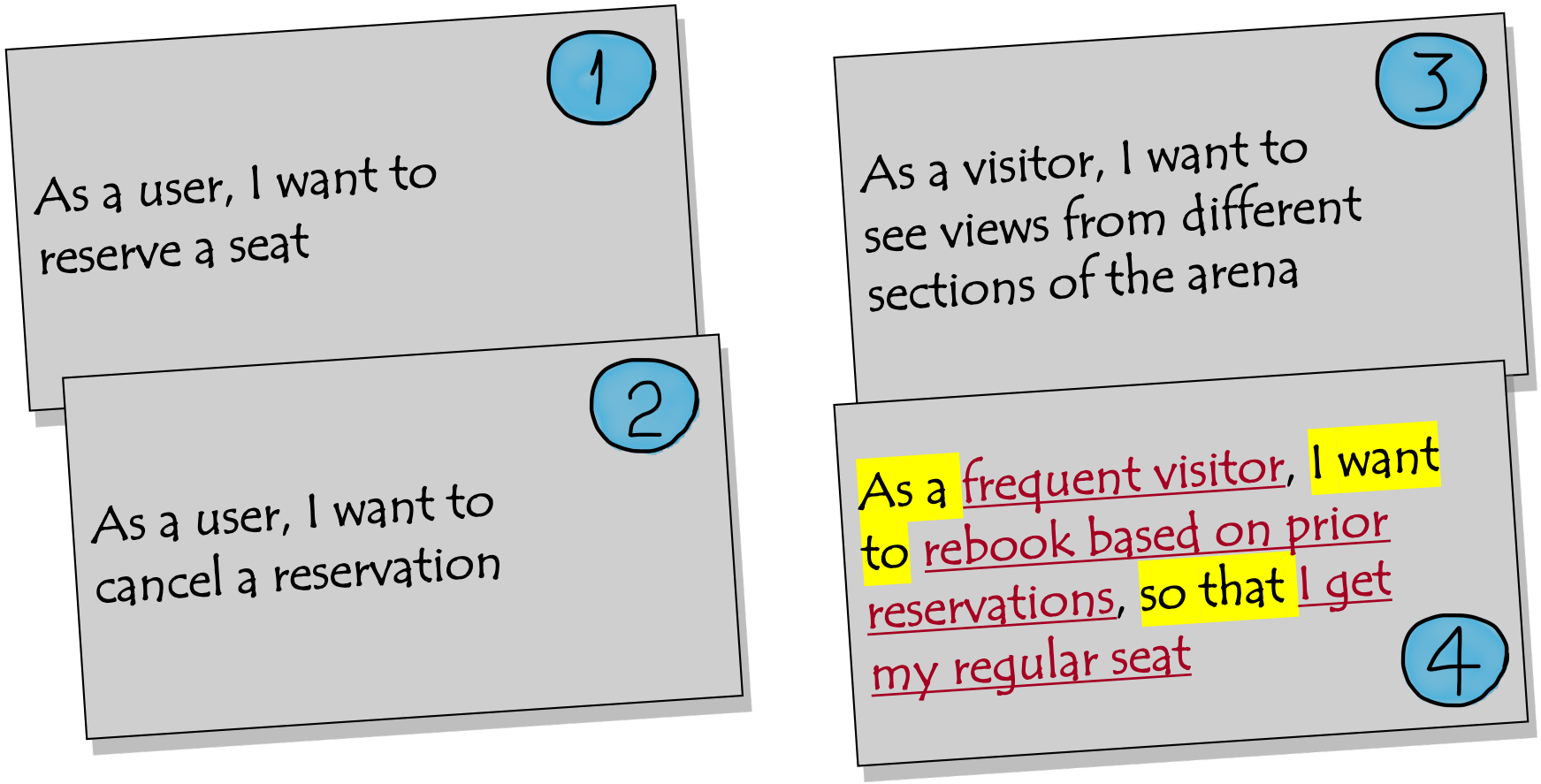
Conversation

1. Details behind the story come out during conversations with Product Owner

Confirmation

1. Acceptance tests confirm that the story was coded correctly

Samples from a reservation web site



What about the details?

”As a user, I can cancel a reservation”

Often it is not possible to start building functionality only based upon the bove information. Still, that is what we often get as a Product Backlog Item (PBI). Therefor we encourage the development team to ask questions during the Sprint Planning and use this as a way to get the details Just In Time.

In this particular case the Dev Team may ask questions like:

- Does the user get a full or partial refund?
  - Is the refund to her credit card or a site credit?
- How far ahead must the reservation be cancelled?
  - Is it the same for all arenas?
  - For all site visitors? Can members cancel later?
- Is a confirmation provided to the user?
  - How?

Different ways of adding details

A. Details as conditions of satisfaction

- essentially tests

B. Details added in smaller sub-stories

Remember:

1. Techniques can be combined
2. These approaches are not mutually exclusive
3. Write items at an appropriate level
4. By the time it's implemented, each item will have conditions of satisfaction associated to it

Product Backlog: Support Communication

- A good Product Backlog supports communication between the Team, the Product Owner and other stakeholders
- Efficient communication happens face-to-face – not in writing
- The form of the Product Backlog Items can support more or less structured communication
- The way the Product Backlog is made has a big influence on the team's velocity

Don’t forget to augment

- User stories do not need to contain everything
- It’s possible to augment written documentation when necessary
- Possible things to augment are:
  - Business rules
  - Data dictionaries
  - User Stories
  - Examples of inputs and expected results
  - Interface descriptions
  - Algoritms

Two parts of a User story

User and *story*

Thinking only in terms of ”the user” may lead to mistakes such as:

- All stories written from the perspective of one user
- Assumes all users have the same goals
- Leads to missing stories and requirements

User role modelling steps

- ☑ Brainstorm an initial set of user roles
- ☐ Organize the original set
- ☐ Consolidate roles
- ☐ Refine roles

Broaden the scope from looking at one user

Allows users to vary based on

- what they will use the software for
- how they use the software
- background
- familiarity with the software/computers

Used in usage-centered design

Advantages of using roles

Users become tangible

- Start thinking of software as solving needs of real people

Avoid saying ”the user”

- Instead we talk about ”a frequent flyer” or a ”repeat traveller”

Incorporate roles into stories

- ”As a <user role>, I want to <goal>
  so that <benefit>

Story-writing workshops

Include

1. Developers
2. Users
3. Customers
4. Others

Brainstorm to generate stories

Goal is to write as many stories as possible

1. Some will be ready to implement
2. Others will be epics

No prioritization

Two common approaches

Start with epics and iterate

Walk through a paper user interface

1. Ask questions as you go
   - What to do next?
   - Possible mistakes here?
   - Possible confusion at this point?
   - Additional information needed?

2. Ask these questions for each user role

Definition of Done (DoD)

- DoD is an agreement within the Scrum Team exactly what it means when the Development Team says “Now we are Done” at the end of the Sprint
- It is good to have a DoD to create clarity and transparency
- The risk of not having a DoD is that we may get misunderstandings, may not meet expectations and may feel that we never done
- There are different ways to come up with a DoD. One way is to understand what is possible to do within each Sprint out of everything we need to do in order to send the product to the customer.
- The perfect DoD is having a potentially releasable product increment at the end of each Sprint
- A minimum DoD is to have done what is necessary in order to meet the acceptance criteria

Release & Long Term Planning

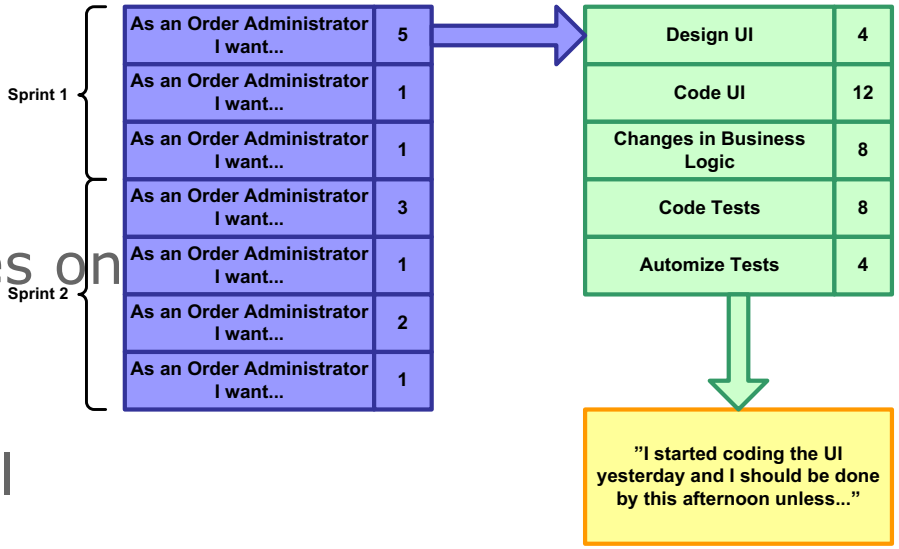
Estimates

What you as Product Owners need to know

1. Estimates should be up to the team
2. Estimates should be created by the team
   - And not by the most senior developer
3. You need estimates before you can prioritize
4. Estimates are not commitments
5. Every estimate should come with a probability
6. Estimates should be of relative, rather than absolute, size

Estimation - Basic Principles

- Three (3) levels of precision
  - Story estimates
  - Task estimates
  - Task remaining work
- Relative (unit less) estimates on story level and above



- Hour estimates on task level
- Estimating is a collaborative effort performed by the whole team
  (e.g. facilitated by using planning poker)
- Estimates are NOT commitments

Always accurate, sometimes precise

Plans must always be accurate

BUT gain in precision over time

1. ”We’ll be done between March and May.”
2. ”We’ll be done sometime in April.”
3. ”We’ll be done on April 24th.”

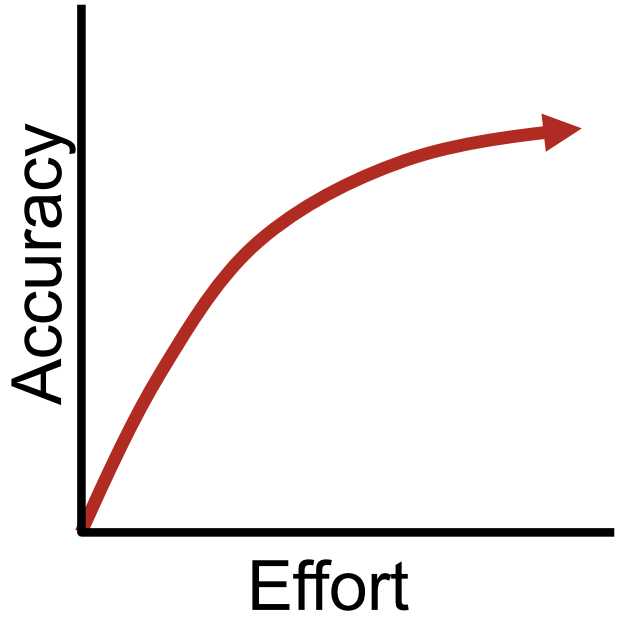
How precisely can you estimate next year’s sales?

1. How precisely do you need to?

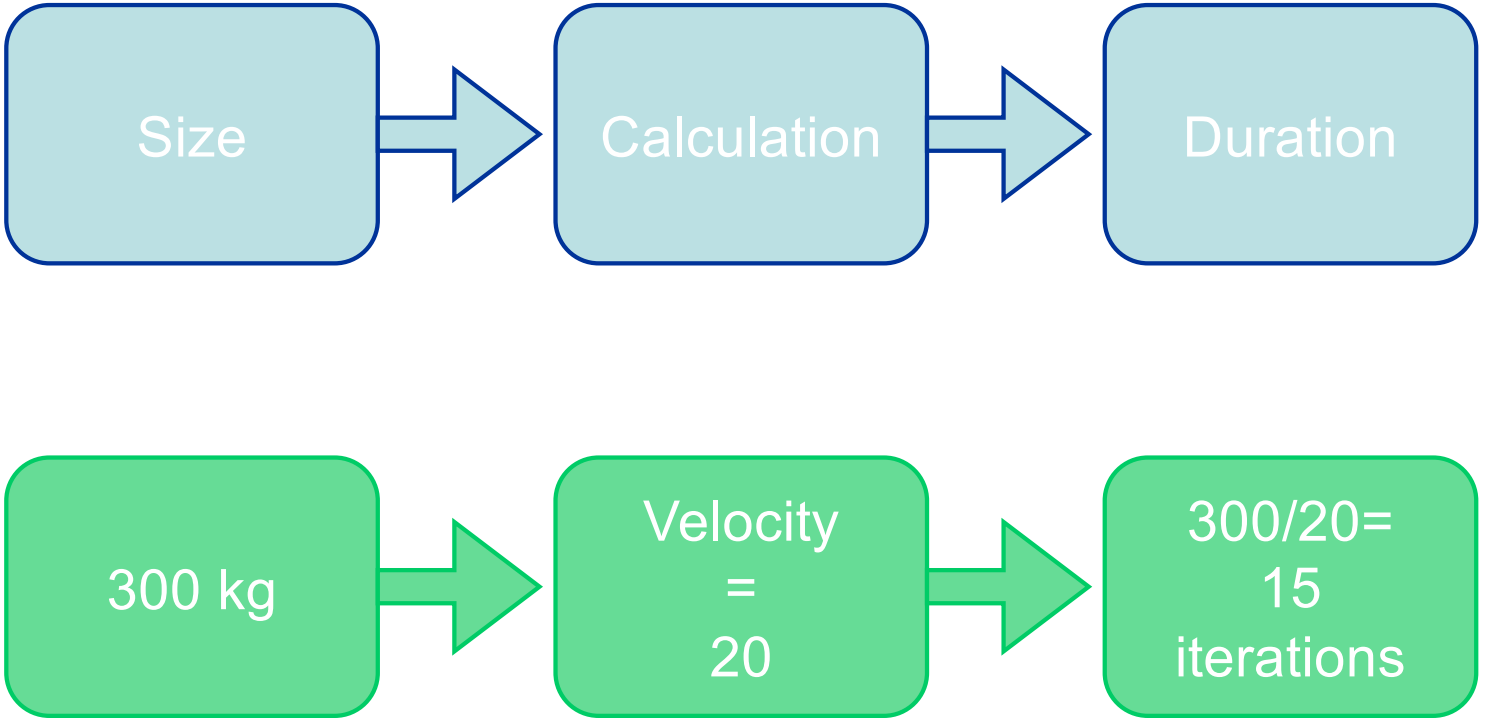
How much effort?

A little effort helps a lot

A lot of effort only helps a little more



Estimate size - derive duration



What's a good plan?

A good plan is one that supports reliable decision-making

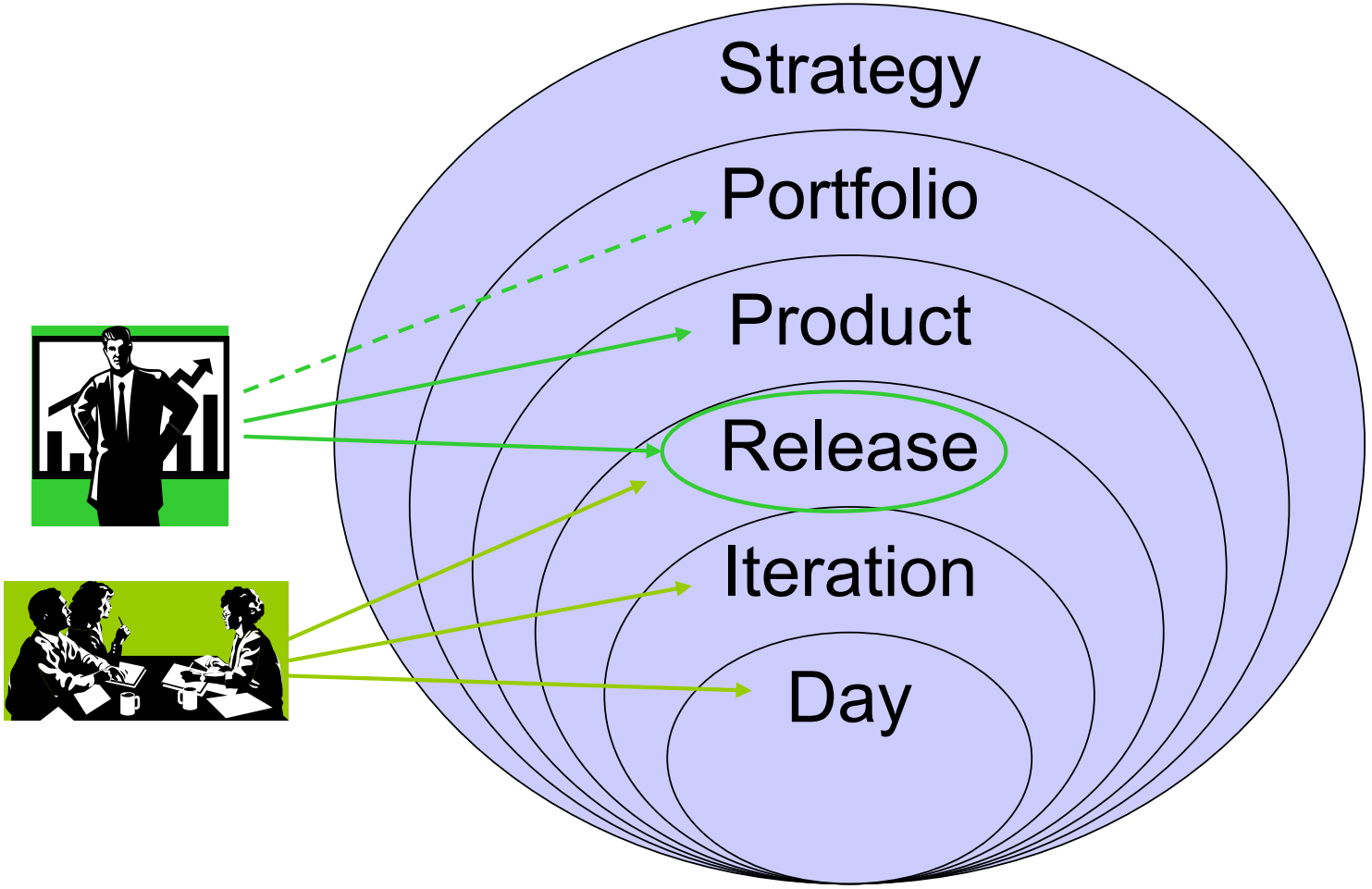
Will go from

1. We'll be done in the fourth quarter
2. We'll be done in November
3. We'll be done November 7th

What makes planning agile?

- Is more focused on planning than the plan
- Encourages change
- Results in plans that are easily changed
- Is spread throughout the project

Planning levels



Release Planning

Purpose

To answer questions such as:

- How much will be done by September 30?
- When can we ship with this set of features?
- How many people or teams should be on this project?

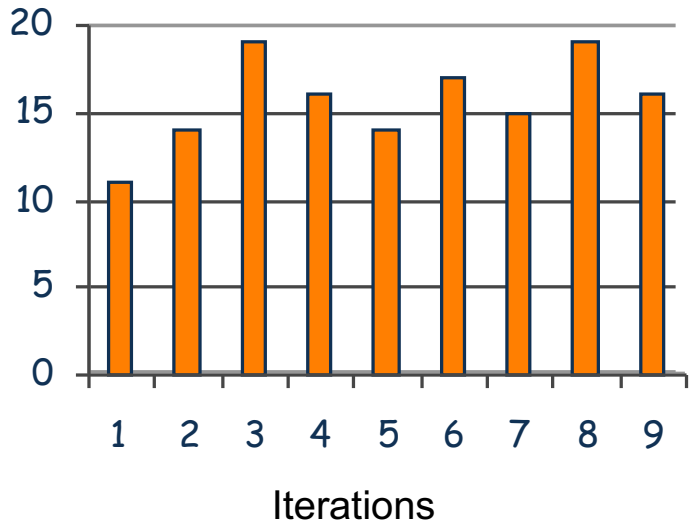
Inputs

- Velocity, i.e. the amount of work completed in a sprint
- Prioritized product backlog

Velocity

A useful long-term measure of the amount of work completed per iteration

Not a prediction of exactly how much work will be completed in *each* iteration



Velocity is measured in the units you use to estimate product backlog items

Different pre-requisites

Fixed-date projects

Fixed-scope projects

Fixed-date planning

How much can I get by <date>?

1. Determine how many sprints you have
2. Measure or estimate velocity as a range
3. Multiply average velocity x number of sprints

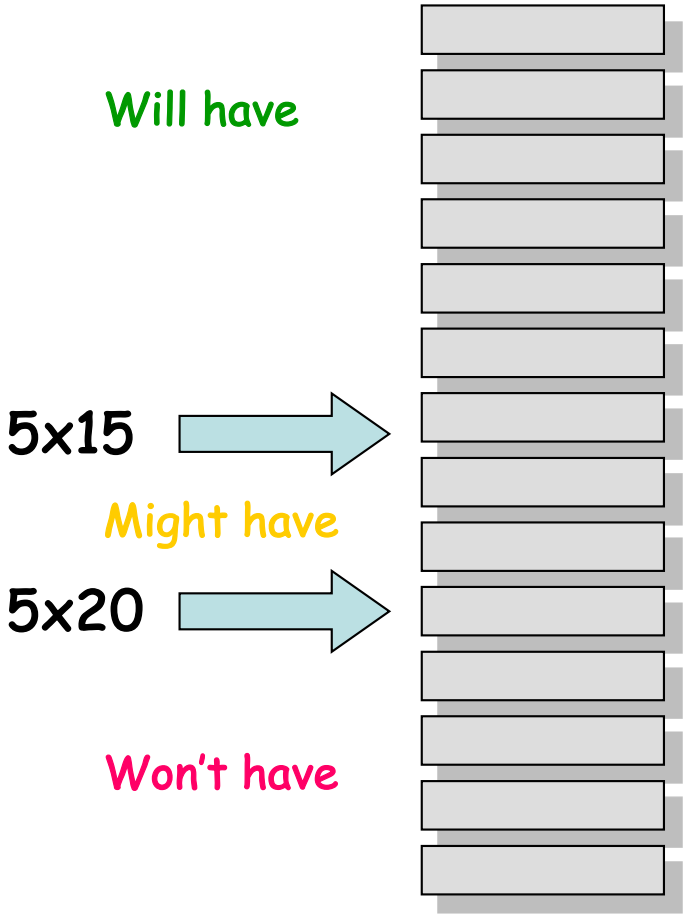
   Count off that many points
   - These are ”Will Have” items

4. Multiply optimistic velocity x number of sprint

   Count off that many more points
   - These are ”Might Have” items

Example of fixed-date planning

Desired release date	1 September
Todays Date	27 March
Number of sprints	5 (monthly)
Average velocity	15
Optimistic velocity	20



Fixed-scope planning

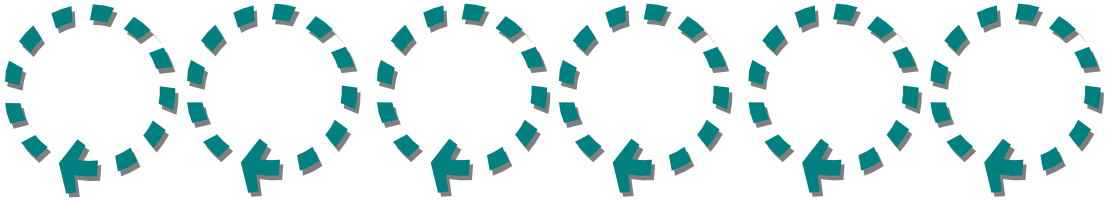
When will all of this be done?

- Sum all the backlog items the customer *needs*
- Measure or estimate velocity as a range
- Divide total story points by optimistic velocity
  1. This is the shortest number of iterations it could take
- Divide total story points by average velocity
  1. This is the ”most” number of iterations it could take

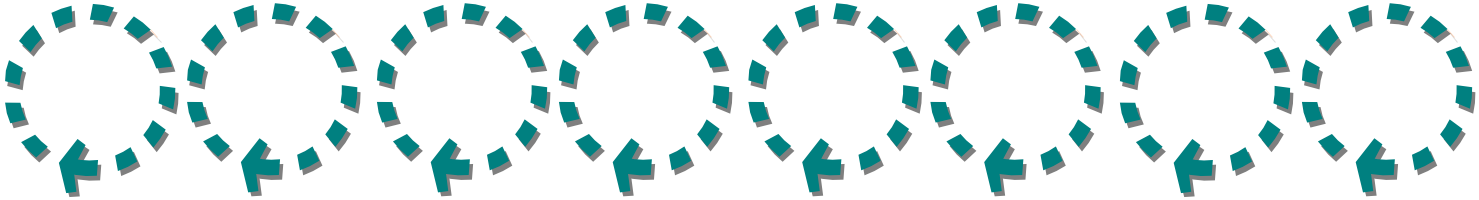
Example of fixed-scope planning

Total story points desired	120
Average velocity	15
Optimistic velocity	20

120 ÷ 20=



120 ÷ 15=



Ranges

Notice in both cases we had a range

Either a scope range:

- ”By that date you’ll have all of these features and some of these.”

Or a date range:

- ”It will take us between 6 and 8 iterations to deliver all of those features.”

Updating the release plan

- Revisit the release plan at the end of every iteration
- Update it based on:
  1. Current understanding of velocity
  2. Current prioritization of the product backlog
- This should be a very short and sweet process

Planning feedback loops