

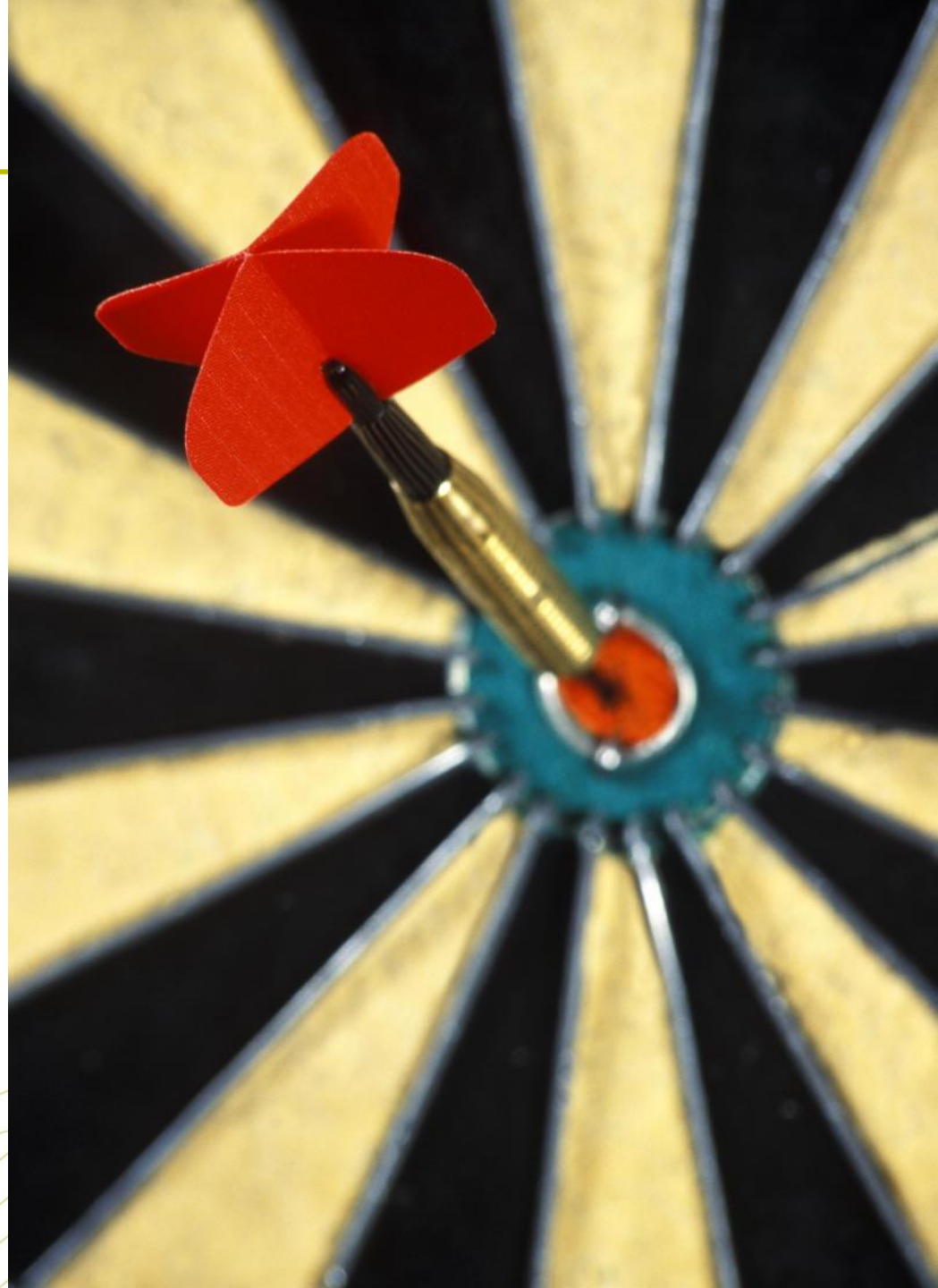


Professional Scrum Master

Welcome to day 1

Course goals

- ⚡ Understand the Scrum process
- ⚡ Understand the roles, events and artifacts of Scrum
- ⚡ Being able to coach on effectiveness, excellence and efficiency
- ⚡ Being able to facilitate inspiring retrospectives
- ⚡ Being able to remove organizational impediments
- ⚡ ... and to become a great Professional Scrum Master



Agenda

	Day 1	Day 2
9.30	Introduction	Recap day 1 & Planning day 2
10.30	Scrum process & Scrum roles	Coaching the Development Team
11.30	Scrum events & Scrum artifacts	Effective retrospectives
12.30	Lunch	Lunch
14.00	Scrum simulation	Coaching the Organization
15.00	Scrum Master profile	Removing impediments
16.00	Coaching the Product Owner	Relation between Agile & Scrum
17.00	Retrospective day 1	Open for questions Retrospective day 2 & course evaluation
18.00		

Getting to know each other

- ⚡ What is your name / role / background?
- ⚡ What are you working on?
- ⚡ What is your background on Agile and / or Scrum?
- ⚡ What is your experience as a Scrum Master
- ⚡ What are your expectations for this course?





Marshmallow Challenge



Marshmallow Challenge

❖ Goal

- ❖ Build the Tallest Freestanding Structure

❖ Building kit

- ❖ 1 meter string (may be cut to pieces)
- ❖ 1 meter masking tape (may be cut to pieces)
- ❖ 20 spaghetti sticks (may be broken to pieces)
- ❖ 1 marshmallow (may not be reduced in size)

❖ Guidelines

- ❖ Measurement will take place 18 minutes after the start
- ❖ The height will be measured from the base till the top of the marshmallow
- ❖ It is not allowed to support the structure with something higher than the base

18:00

守 破 離

Shu Ha Ri



Scrum

Reminder of project objectives

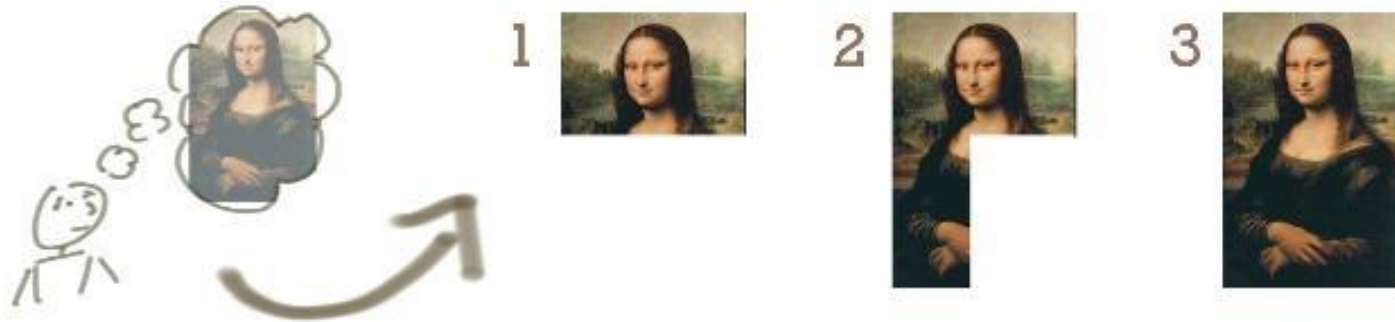
- Provide a solution to a business need
 - Answer as fast as possible to this need
 - Check throughout the project life, the alignment with business objective
 - Deliver a good quality product
 - Meet specific constraints (time, cost)
-
- The fact: Standard methodologies provide a large number of features that are not used





Provide a working tool as early as possible

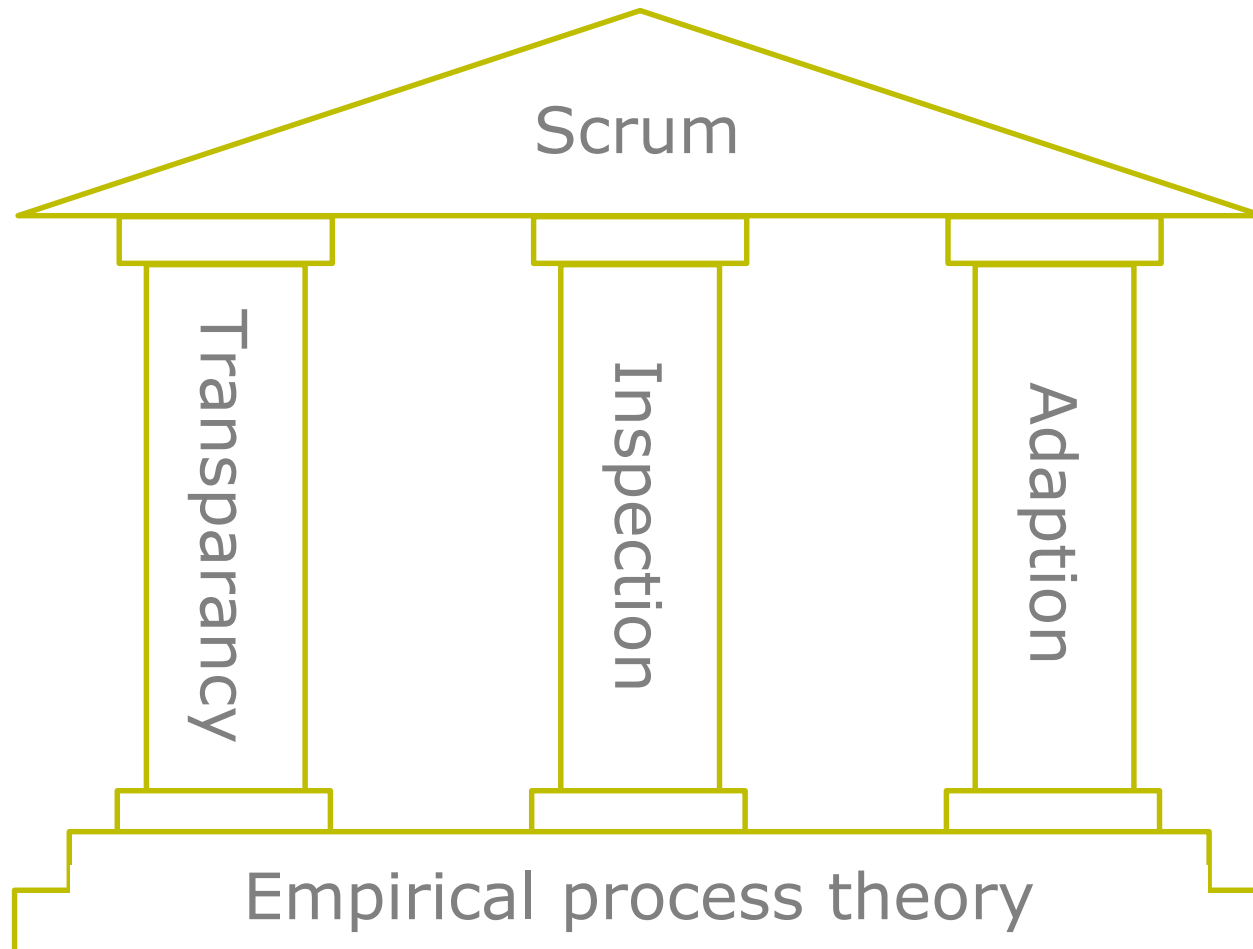
Iterate & increment



*Scrum is not a process or a technique for building products;
rather, it is a framework within which you can
employ various processes and techniques.*

A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

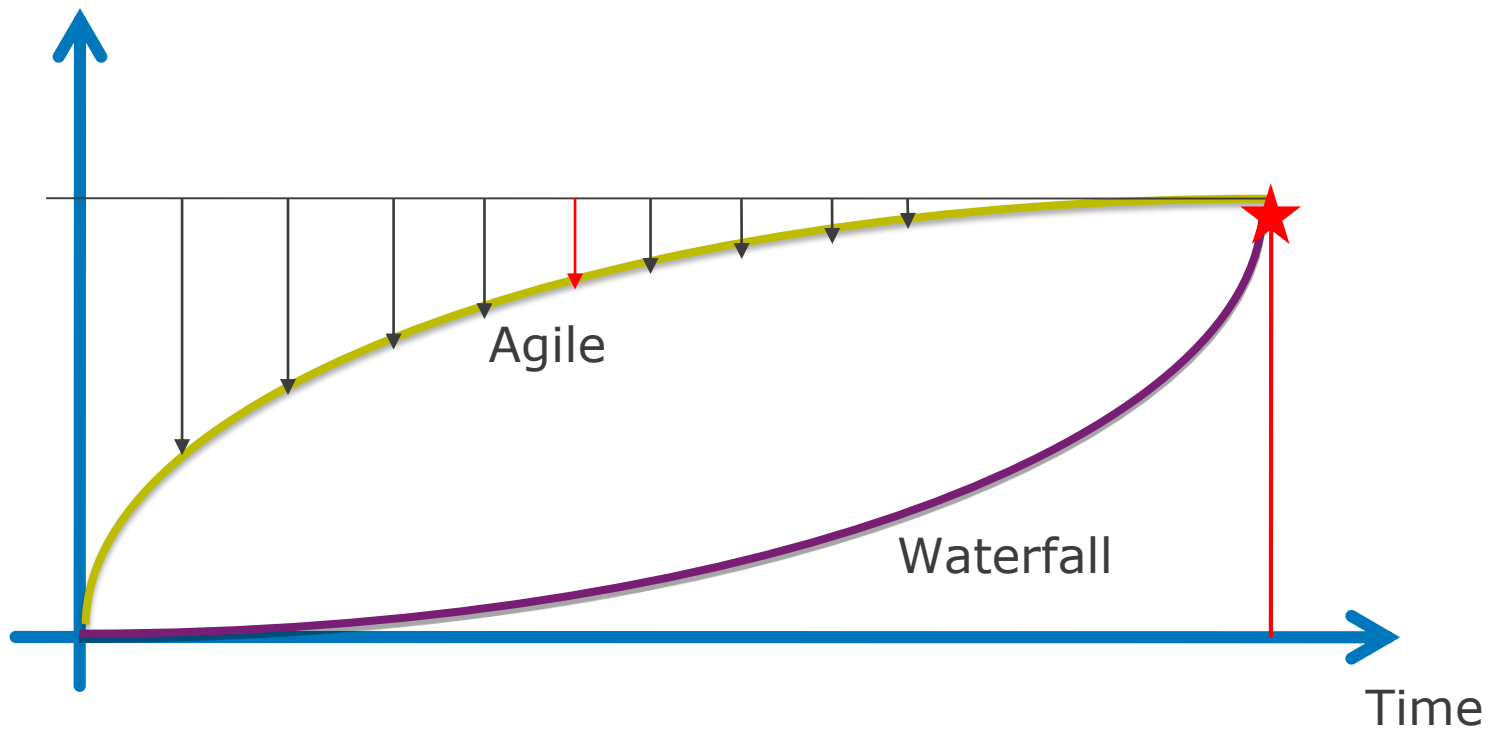
Jeff Sutherland, Ken Schwaber, 1995



Elevator pitch of Scrum

- ⚡ Scrum is an Agile process that allows us to focus on delivering the highest business value in the shortest time.
- ⚡ It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).
- ⚡ The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- ⚡ Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

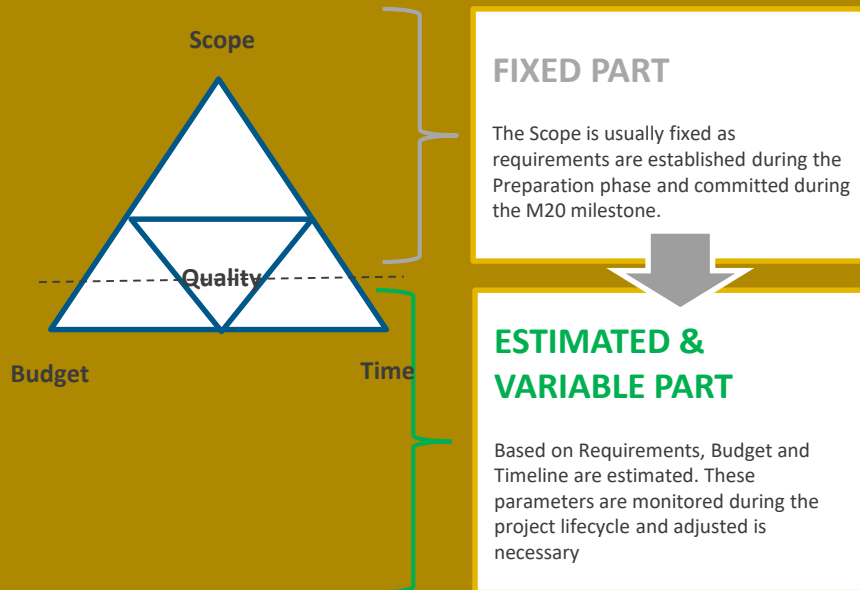
Business Value



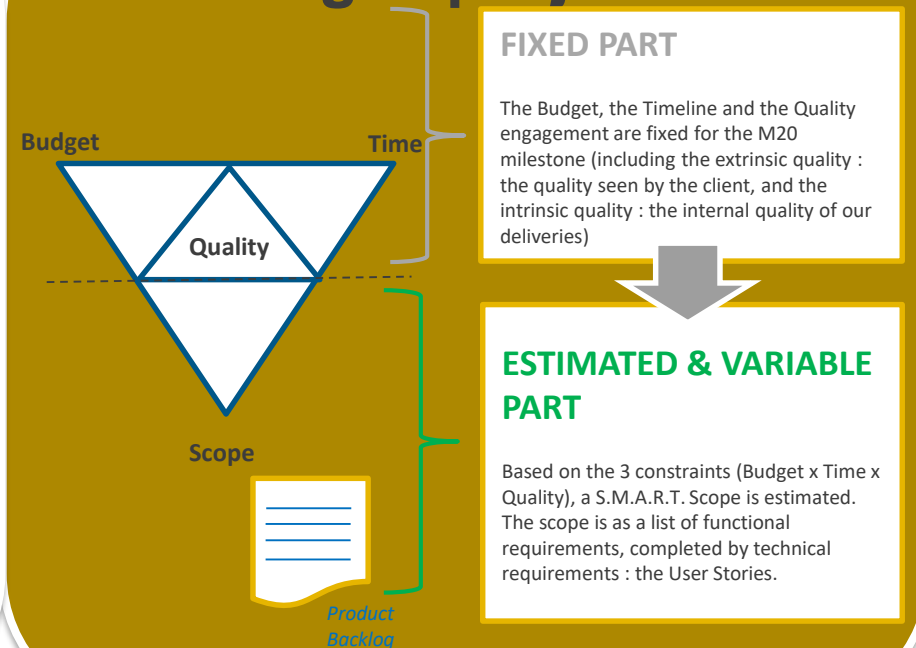
Business constraints

- ✦ In most cases, Agile projects have a fixed time and budget, and adapt the scope

Waterfall project



Agile project



- ✦ Usually, Waterfall projects have a fixed scope, and adapt the time and budget to deliver this scope

The ... 'relay race' approach to product development ... may conflict with the goals of maximum speed and flexibility.

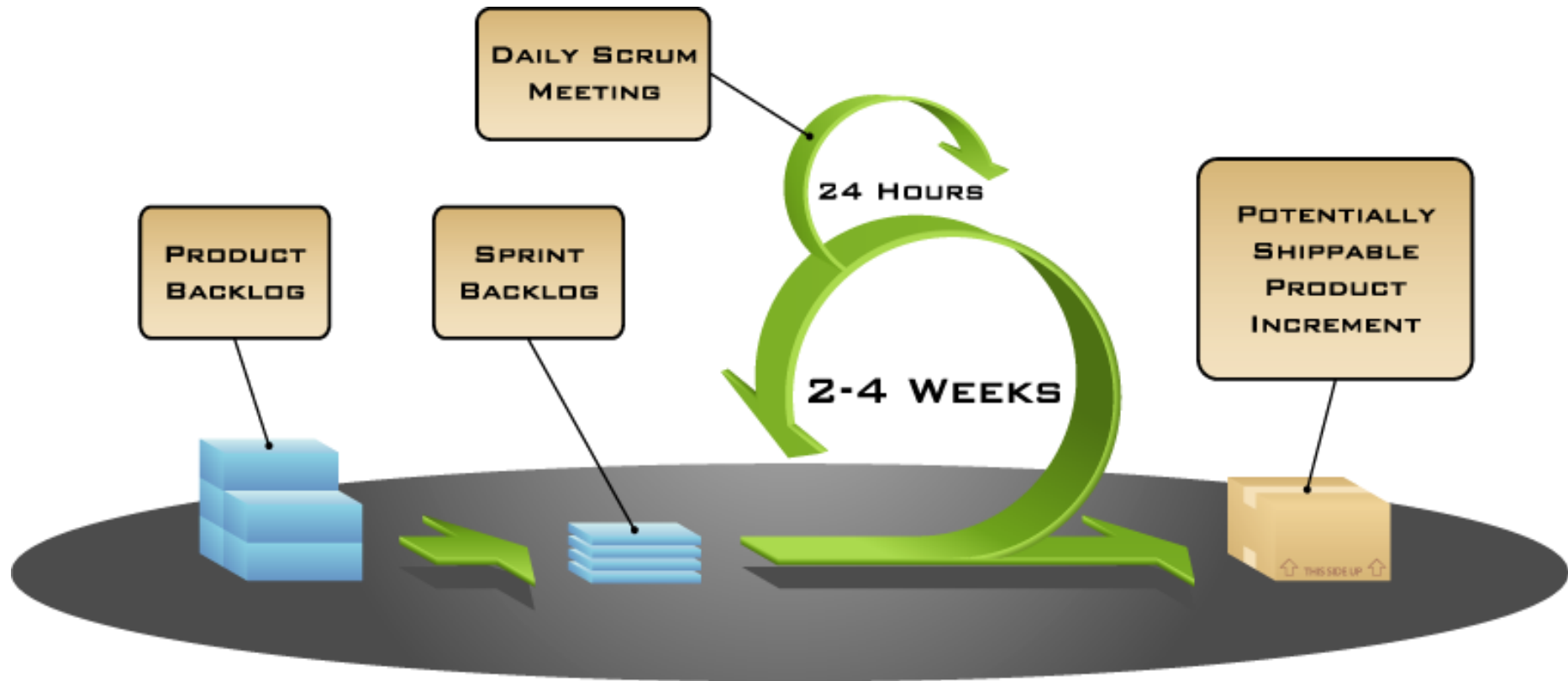
Instead a holistic or 'rugby' approach – where a team tries to go the distance as a unit, passing the ball back and forth – may better serve today's competitive requirements



Characteristics of Scrum

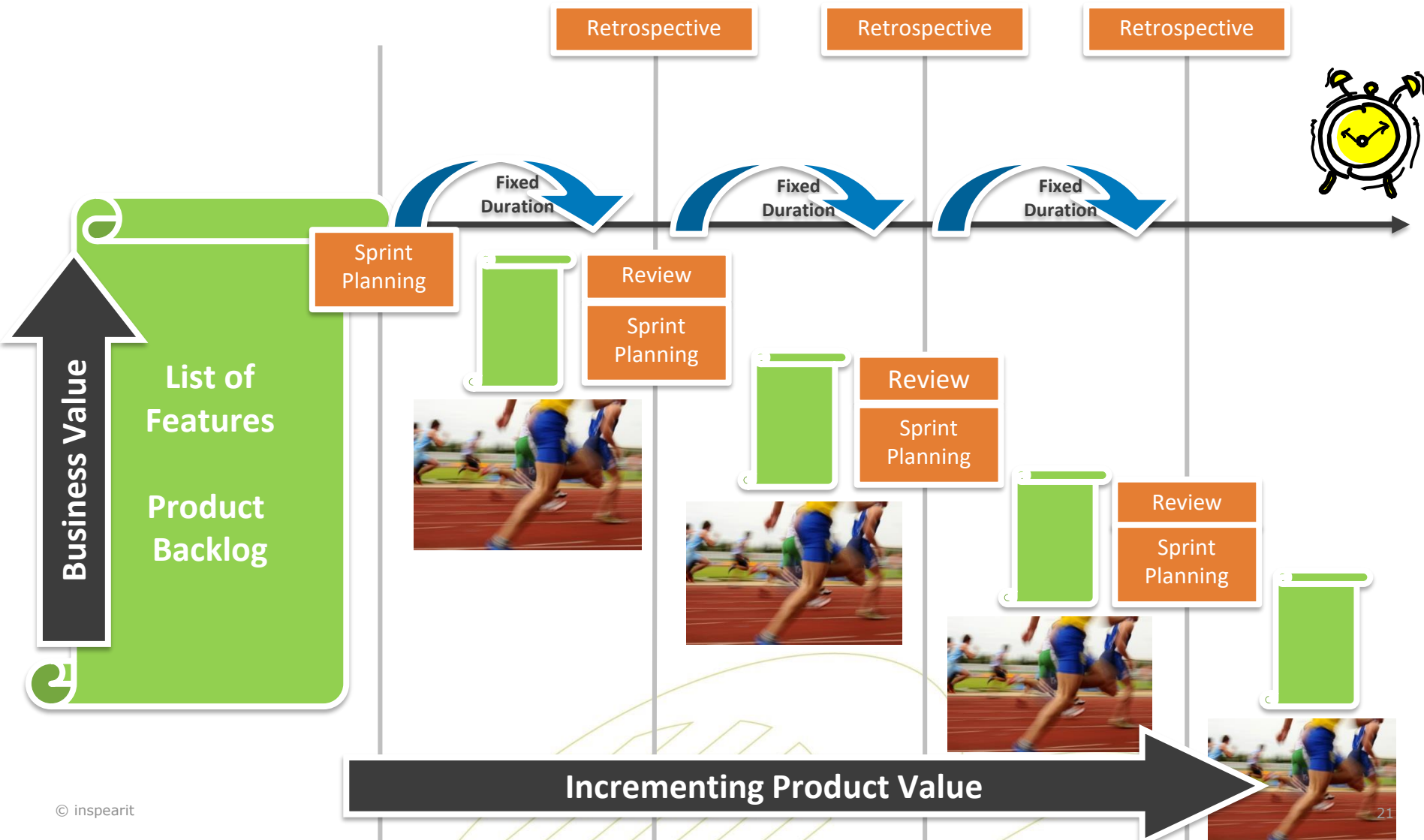
- ✦ Self-Organization
 - ✦ Work in your comfort zone
- ✦ No single point of control
 - ✦ Command and control based on cooperation
- ✦ Interdisciplinary teams
 - ✦ No isolated activities or lack of transparency
- ✦ Emergent behavior
 - ✦ Help each other. A chain is as weak as the weakest link
- ✦ Outcomes emerge in context
 - ✦ Empirical, realistic progress
- ✦ Team performance far greater than sum of individuals
 - ✦ Together Each Achieve More

Scrum process



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Scrum process



Scrum does NOT cover all of...

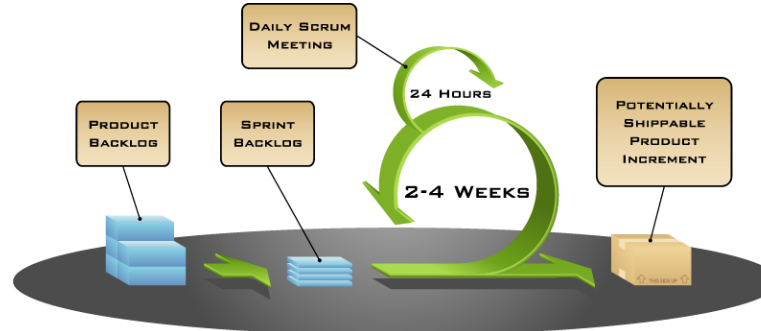
Configuration Management

Requirements Development

Change Management

Deployment Management

Outsourcing



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Document Management

Offshoring

Reporting PMO

Measurements & Analysis

Quality Assurance

Scrum framework

Roles

- Product Owner
- Scrum Master
- Development Team

Events

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Artifacts

- Product Backlog
- Sprint Backlog
- Increment

Events

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

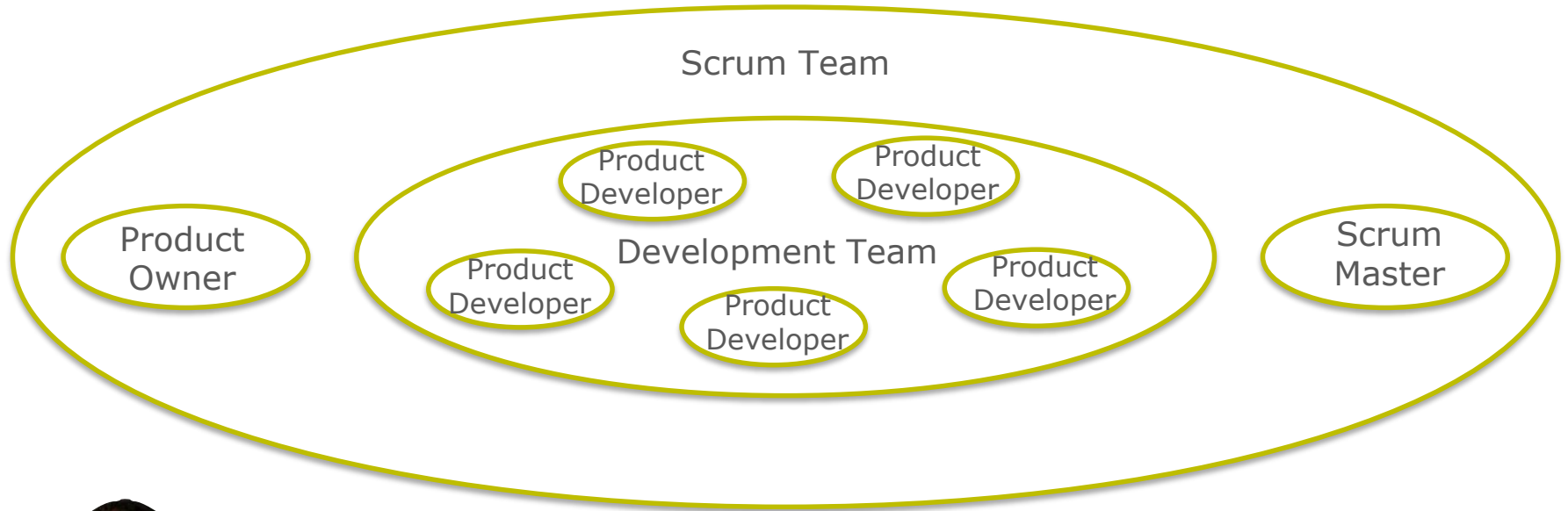
Artifacts

- Product Backlog
- Sprint Backlog
- Increment

Roles

- Product Owner
- Scrum Master
- Development Team

Scrum Team



Product Owner

- ✦ Represents all stakeholders
- ✦ Decides where the team should go
 - ✦ Not how to get there
 - ✦ Not their speed
- ✦ Defines scope / vision / roadmap
- ✦ Prioritizes
- ✦ Owns Product Backlog
- ✦ Does NOT estimate stories

- ✦ *Usually NOT the line manager*



Scrum Master

- ⚡ Enforces Scrum practices
- ⚡ Removes impediments
- ⚡ Coaching rather than command & control
- ⚡ Usually part of the team

- ⚡ *Usually NOT the tech guru*
- ⚡ *Usually NOT the line manager*



Development Team

- ⚡ Typically 3 - 9 people
- ⚡ Cross-functional
 - ⚡ Programmers, testers, analysts, user experience designers, etc.
- ⚡ Members should be full-time
 - ⚡ May be exceptions (e.g. DBA's)
- ⚡ Teams are self-organizing
 - ⚡ Ideally, no titles but rarely a possibility
- ⚡ Membership should change only between sprints



Scrum framework

Events

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Artifacts

- Product Backlog
- Sprint Backlog
- Increment

Roles

- Product Owner
- Scrum Master
- Development Team

Artifacts

- Product Backlog
- Sprint Backlog
- Increment

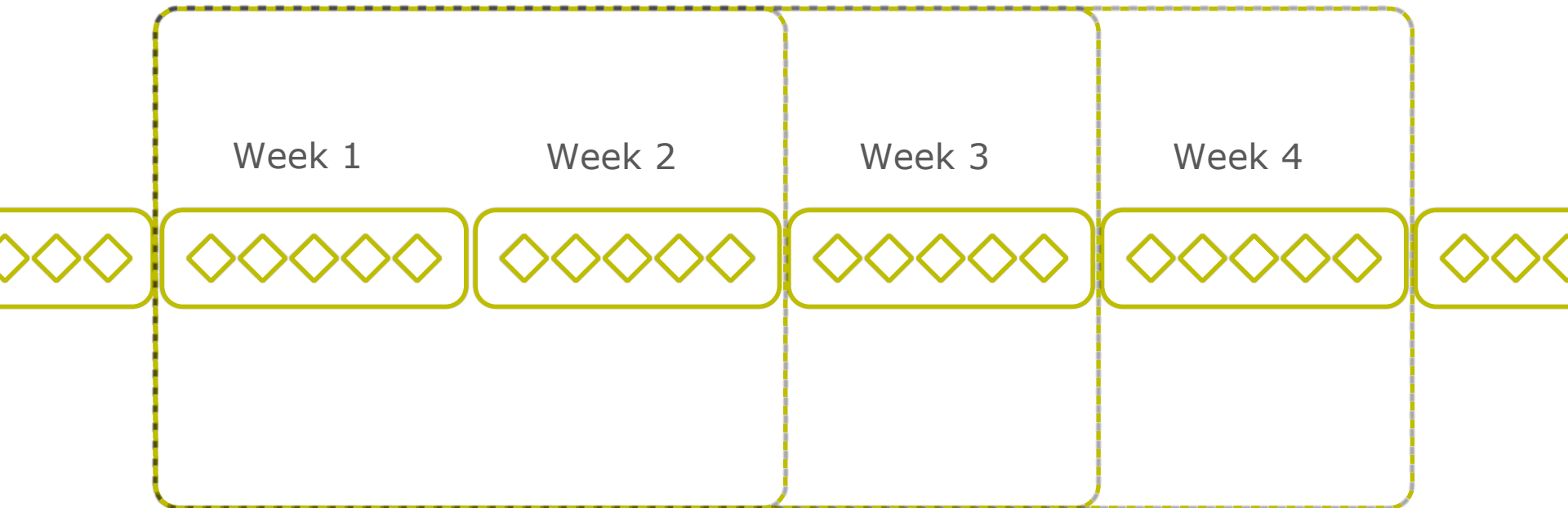
Roles

- Product Owner
- Scrum Master
- Development Team

Events

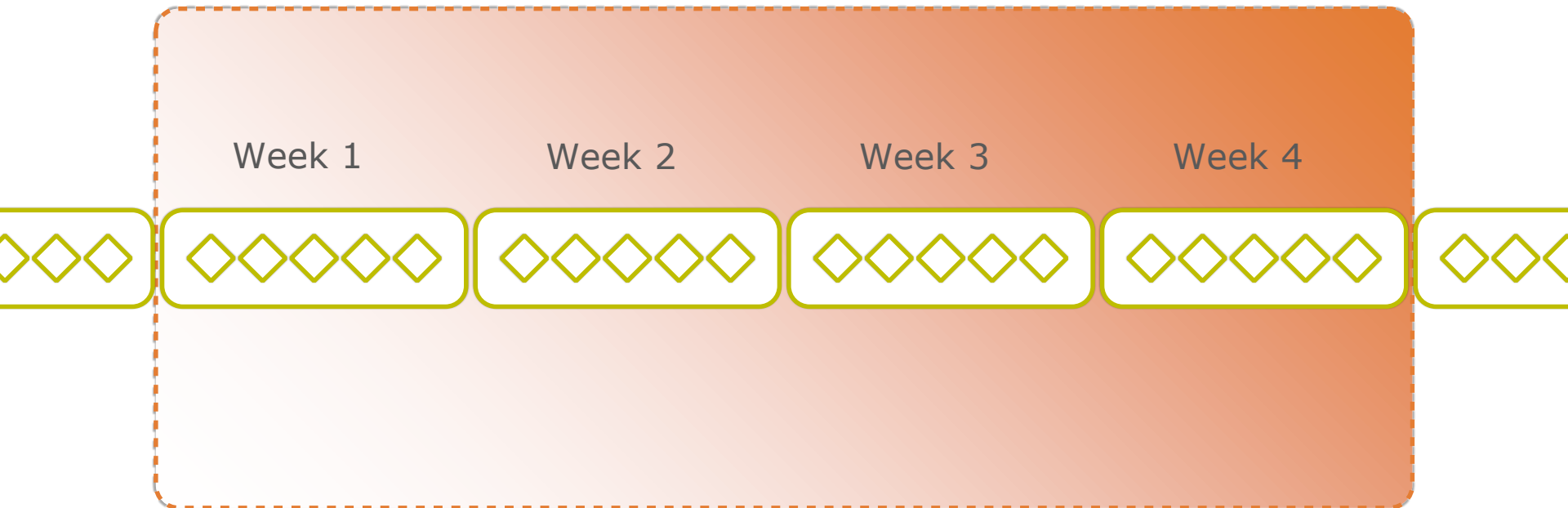
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Time box of Scrum



Sprint

Time box: max. 1 month



- ⚡ Limited period of time: it is the time required to:
 - ⚡ Build a set of features required by the Product Owner
 - ⚡ Deliver this set of features with a shippable status
- ⚡ The team can rely on external help
- ⚡ Nobody gives directives or instructions to the team
- ⚡ The iteration backlog is frozen
- ⚡ Iteration can be abnormally terminated if it is not viable
 - ⚡ By the Scrum Master
 - ⚡ On the request of the Product Owner
 - ⚡ On the request of the team
- ⚡ The scope should be reviewed and exceptionally modified
- ⚡ The team obligations are to:
 - ⚡ Attend daily Scrum meetings
 - ⚡ Maintain the iteration backlog updated and make it visible

- ⚡ Scrum projects make progress in a series of Sprints
- ⚡ A Sprint is
 - ⚡ a container for Scrum events
 - ⚡ A time-box of one month or less
- ⚡ Each Sprint may be considered a project with no more than a one-month horizon
- ⚡ A constant duration leads to a better rhythm
- ⚡ A product is designed, coded, tested and documented during the sprint

No changes during a sprint



Plan sprint durations around how long you can commit to keeping change out of the sprint

Sequential versus overlapping



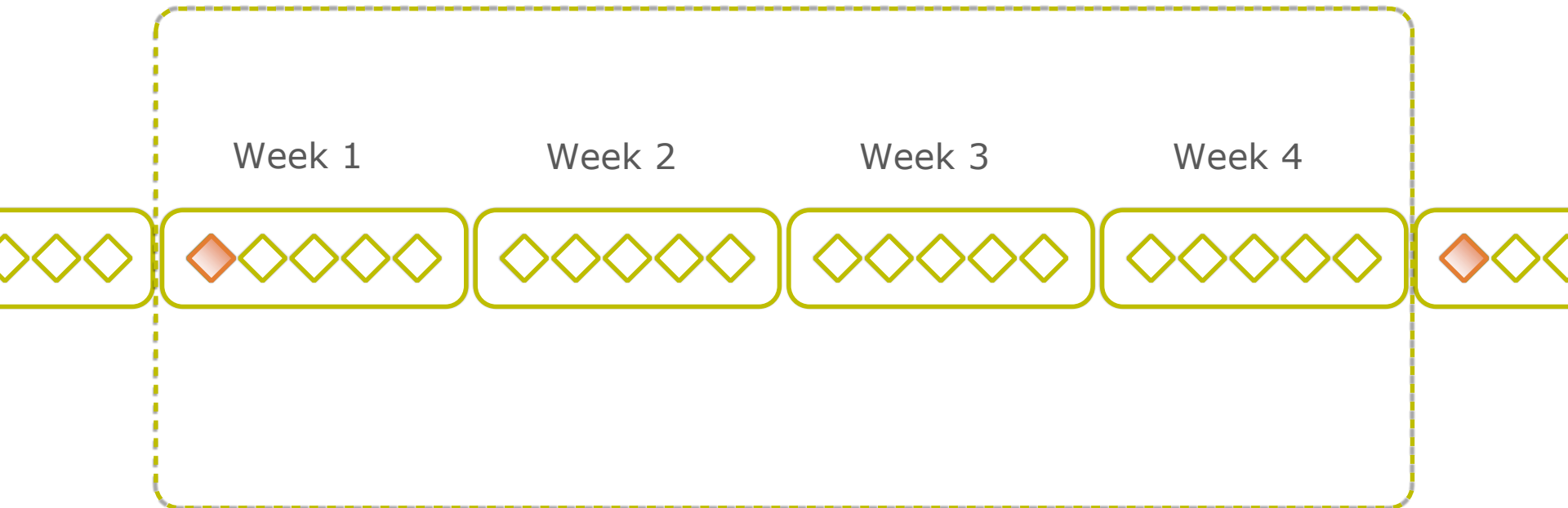
Rather than doing all of one thing at a time ...

... Scrum teams do a little of everything all the time



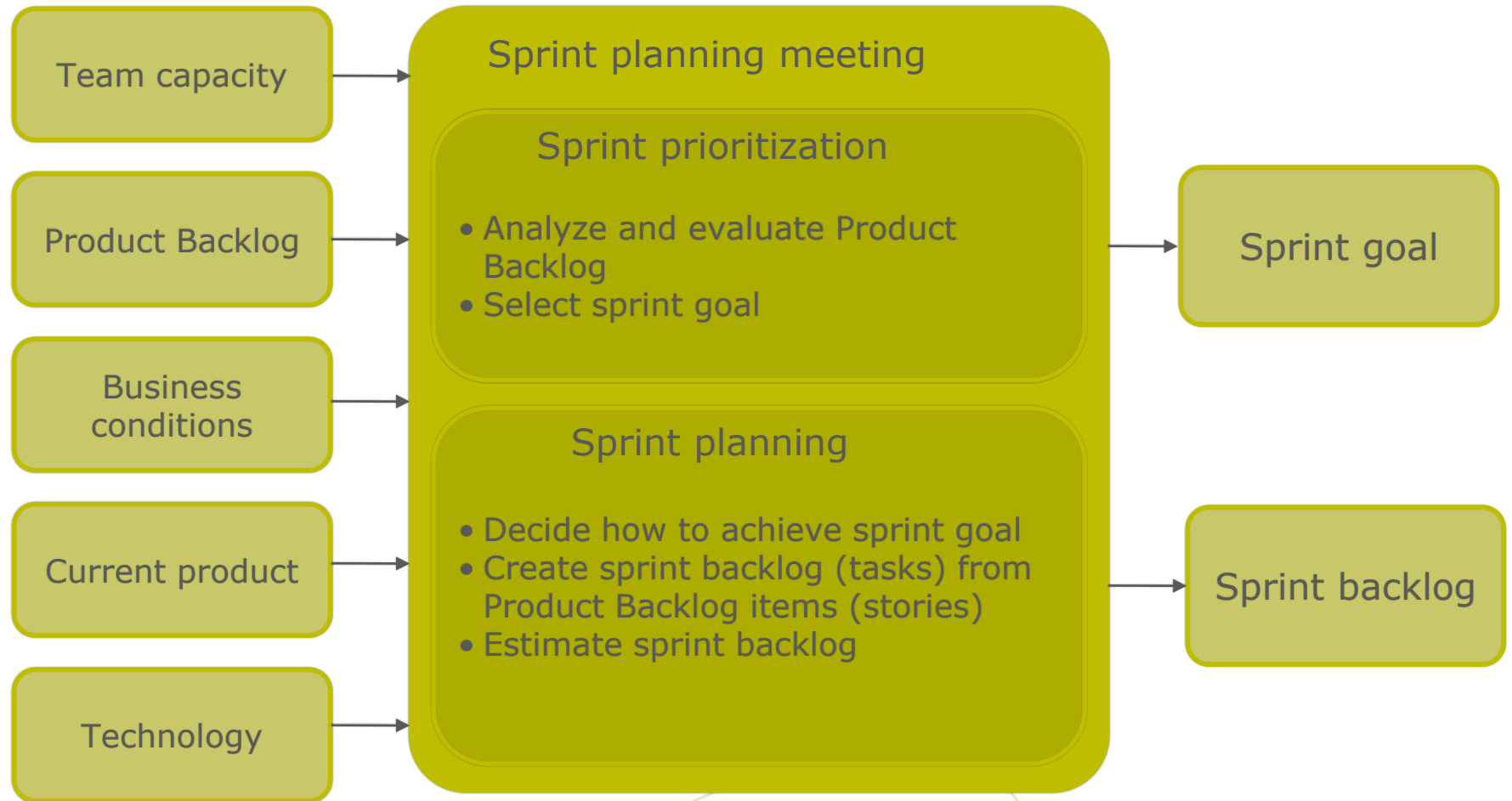
Sprint planning

Time box: max. 8 hours



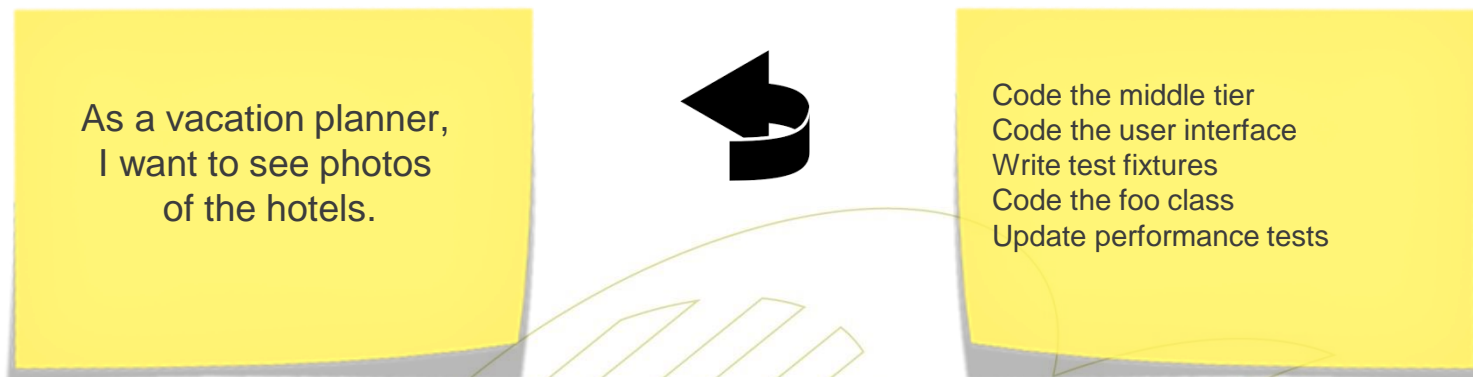
Sprint planning

- ✦ Planning of all work for the next Sprint
- ✦ The whole Scrum Team is involved
- ✦ Two important questions:
 - ✦ **What** can be delivered as an Increment at the end of the Sprint?
 - ✦ **How** do we accomplish the necessary work for this Increment?
- ✦ Establishing a Sprint Goal for this Sprint
 - ✦ Gives meaning to the Increment in progress



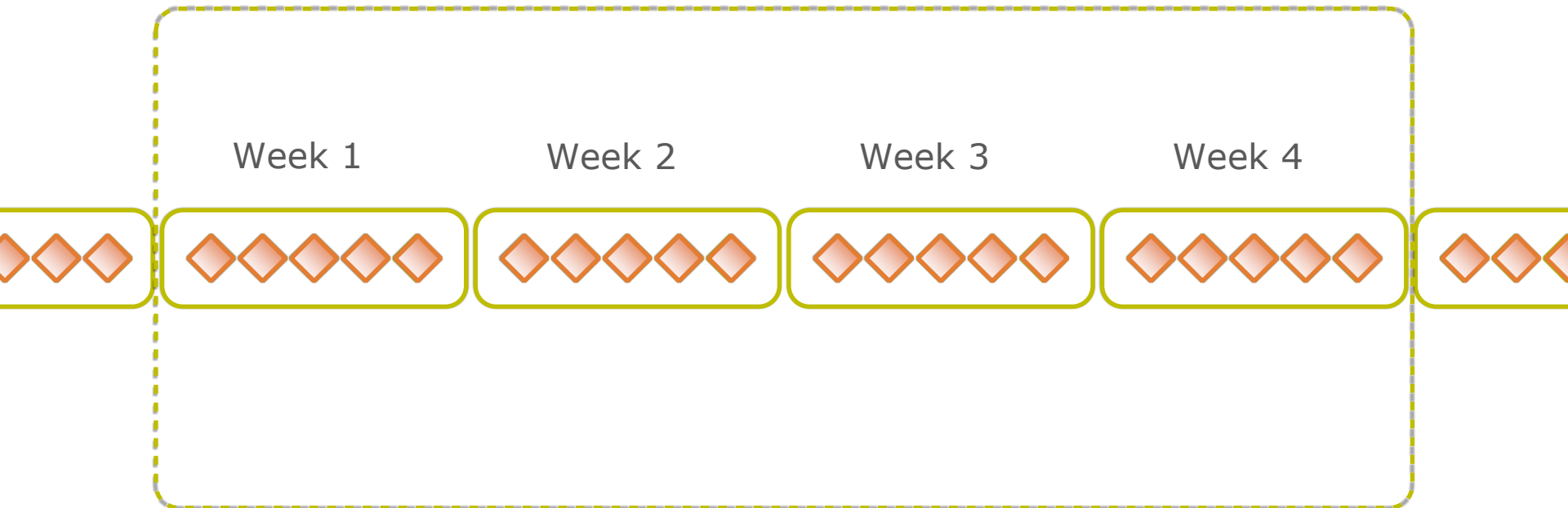
Sprint planning

- Team selects items from the Product Backlog they can commit to completing
- Sprint backlog is created
 - Tasks are identified and each is estimated
 - Collaboratively, not done alone by the Scrum Master
- High-level design is considered



Daily Scrum

Time box: max. 15 minutes



What did you do yesterday (for the Sprint Goal) ?

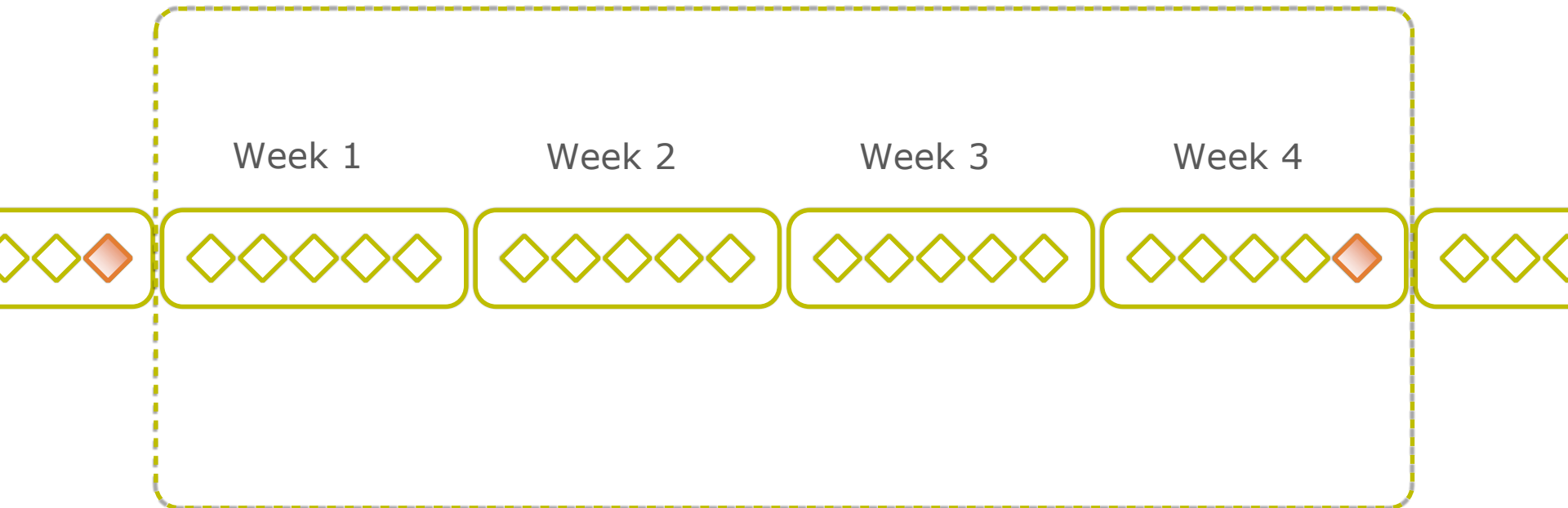
What will you do today (for the Sprint Goal) ?

Is anything in your way (to reach the Sprint Goal)?

- ⚡ These are not status updates for the Scrum Master
- ⚡ They are commitments in front of peers

Sprint review

Time box: max. 4 hours

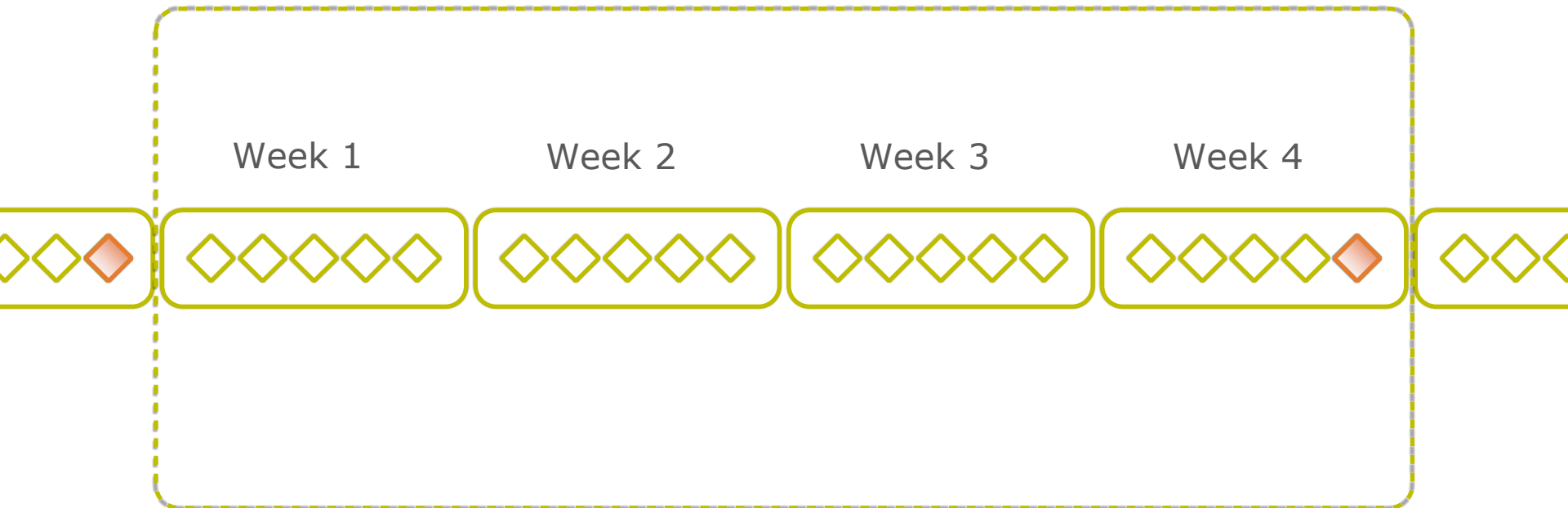


Sprint review

- ⚡ Goal
 - ⚡ Inspection of the delivered Increment
 - ⚡ Adapt the Product Backlog on current insights
- ⚡ Identification of backlog items completed
- ⚡ Demonstration of the completed work
- ⚡ Outlook on the next Sprint
 - ⚡ Necessary input for the Sprint planning
- ⚡ Informal meeting between Scrum Team and stakeholders
- ⚡ Result
 - ⚡ Revised Product Backlog

Sprint retrospective

Time box: max. 3 hours



Sprint retrospective

- ⚡ Goal
 - ⚡ Inspection and adaption of people, relations, processes and tools
- ⚡ Many techniques and styles available
- ⚡ After Sprint Review but before Sprint Planning
- ⚡ Review of the Definition of Done
- ⚡ Result
 - ⚡ Plan to improve as a team in the next Sprint

Artifacts

- Product Backlog
- Sprint Backlog
- Increment

Roles

- Product Owner
- Scrum Master
- Development Team

Events

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Roles

- Product Owner
- Scrum Master
- Development Team

Events

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Artifacts

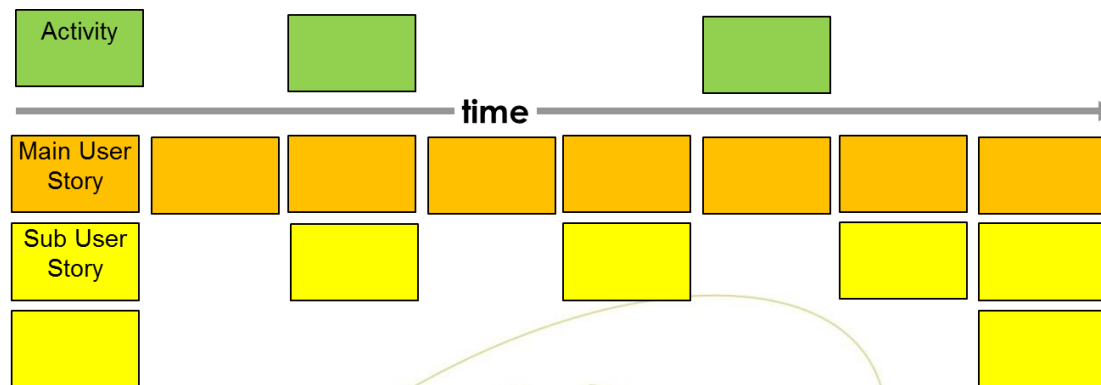
- Product Backlog
- Sprint Backlog
- Increment

Product Backlog

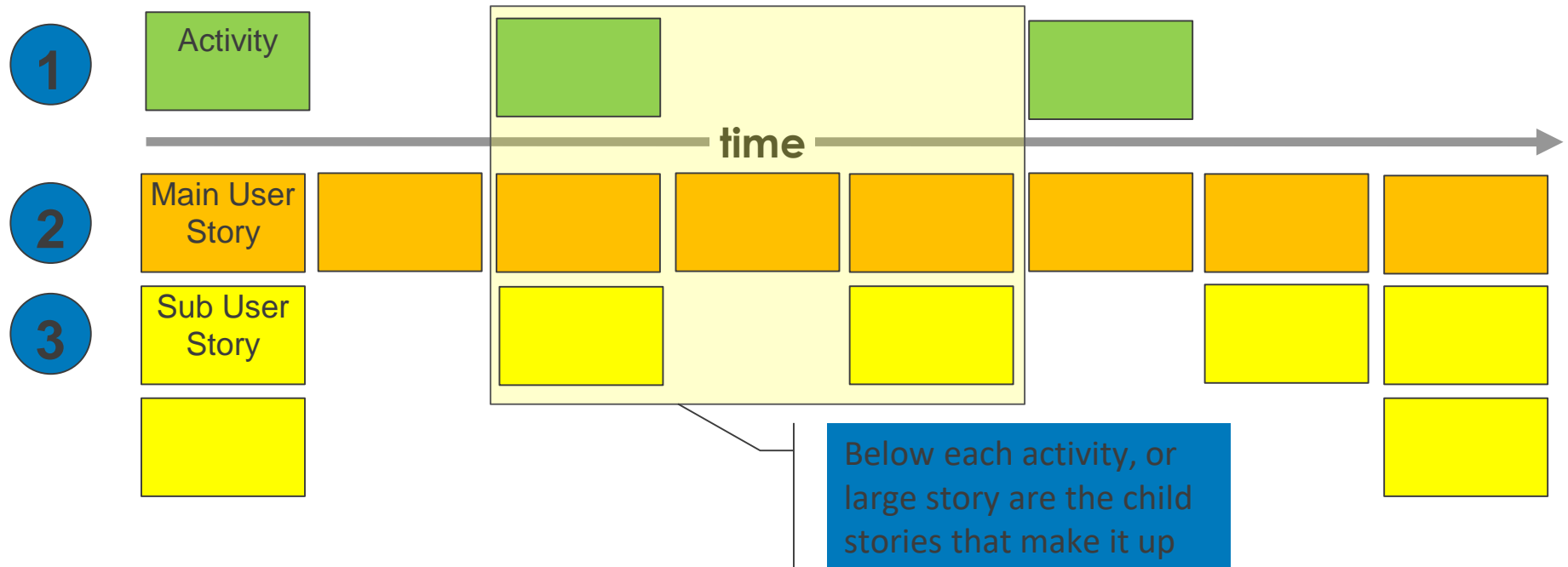
- ⚡ An ordered and prioritized list of everything that might be needed in the product
 - ⚡ features, functions and requirements
 - ⚡ enhancements, improvements and fixes
- ⚡ Product Backlog items have the attributes of a description, order, estimate and value.
- ⚡ The single source of requirements for any changes to be made to the product
- ⚡ Owned by the Product Owner

Story mapping

- ⚡ User Story Mapping is an approach to Organizing and Prioritizing user stories
 - ⚡ Identify the workflow or value chain
 - ⚡ Show the relationships between activities, tasks, sub-tasks
 - ⚡ Validate the completeness of your backlog
 - ⚡ Help prioritize the user Stories and plan releases



Story mapping



- 1 Start with big users activities to tell the big story of the product
- 2 Add The workflow of task-centric stories under each activities
- 3 Add other tasks that may be done by the users

Sprint Backlog

- ⚡ The Sprint Backlog is the set of
 - ⚡ Product Backlog items selected for the Sprint
 - ⚡ Plan for delivering the product Increment
 - ⚡ Plan for realizing the Sprint Goal
- ⚡ Visualizes all of the work that has been identified
- ⚡ Visualizes the progress of the work items

Increment

- ⚡ The Increment is the
 - ⚡ sum of all the Product Backlog items completed during a Sprint and
 - ⚡ the value of the increments of all previous Sprints
- ⚡ At the end of a Sprint, the Increment must be “Done”
 - ⚡ in useable condition
 - ⚡ meets the Scrum Team’s definition of “Done”

Definition of Done (DoD)

- ⚡ Common understanding when parts of the Increment are actually completed
- ⚡ Different DoD's per Scrum Team
- ⚡ Stimulates transparency
- ⚡ DoD determines how many Product Backlog items can be selected for a Sprint
- ⚡ Improves over time
 - ⚡ More specified definition of quality



Scrum simulation



Develop an mobile application

⚡ Timing :

- ⚡ 10mn to select the project and define the vision
- ⚡ 20mn to create a story mapping
- ⚡ 3 iterations of 20mn

⚡ Roles :

- ⚡ Who are the Scrum Master and the Product Owner ?
- ⚡ What are their roles ?

Develop an mobile application

✦ Exercise :

- ✦ Define the vision : who are the users? what are they expecting? What is the issue to solve? What are the 3 main functionalities? (10mn)
- ✦ Story mapping : define and prioritize user stories (20mn)
- ✦ 3 iterations :
 - Sprint planning : 5mn
 - Development (drawing of the screen) : 10mn
 - Sprint review : 2mn
 - Retrospective : 3 mn

✦ End of the game :

- ✦ Each team show their product (the drawings)
- ✦ We perform a global debrief

Develop an mobile application

✦ Ideas :

- ✦ SafeCard: credit card management application for smartphone
- ✦ Restaurant&Friends: organize lunch with colleagues
- ✦ VeggiesShare: exchange vegetable that you produce with other gardeners
- ✦ Poker&Co: Online poker game

✦ Or your own idea!

Scrum Master profile

Characteristics of a good Scrum Master

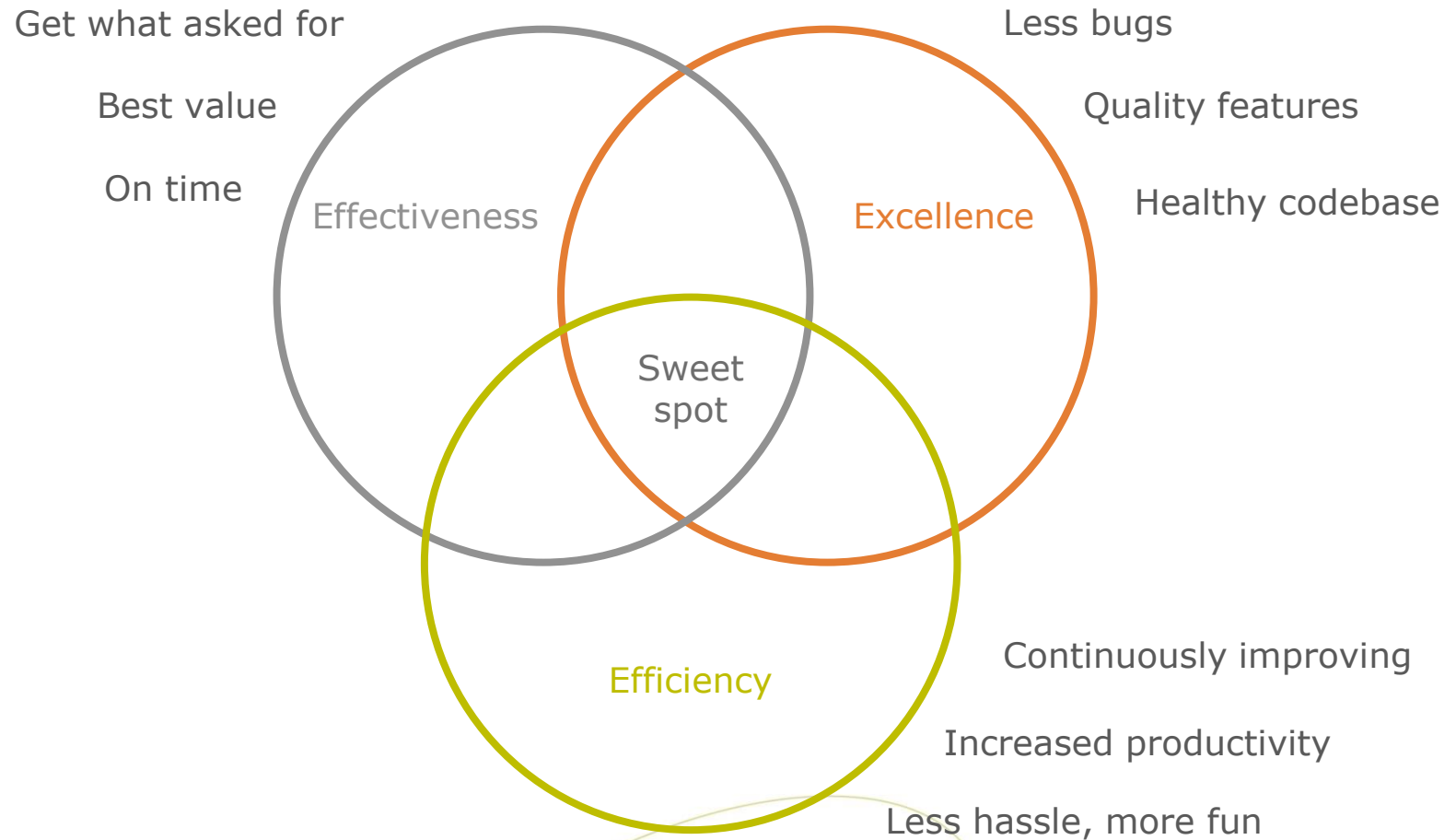
- ✦ Responsible
 - ✦ assume responsibility for the team's adoption of Scrum and practice of it.
- ✦ Humble
 - ✦ willing to do whatever is necessary to help the team achieve its goal.
- ✦ Collaborative
 - ✦ work to ensure a collaborative culture exists within the team.
- ✦ Committed
 - ✦ feel the same high level of commitment to the project and the goals of the current sprint as do team members.

Characteristics of a good Scrum Master

- ✦ Influential
 - ✦ know how decisions are made in the organization, who makes them
- ✦ Knowledgeable
 - ✦ need to be conversant around the key (technical) issues.

Coaching styles

- ✦ Transformational
 - ✦ Reflective / mirroring / questioning
- ✦ Solution oriented
 - ✦ Focus on things that work instead of the problem
- ✦ Provocative
 - ✦ Confront and stimulate
- ✦ Mentoring
 - ✦ Share your experience and translate to their situation
- ✦ Teaching
 - ✦ Share knowledge and insights
- ✦ Result oriented
 - ✦ Practical applications and guide towards results



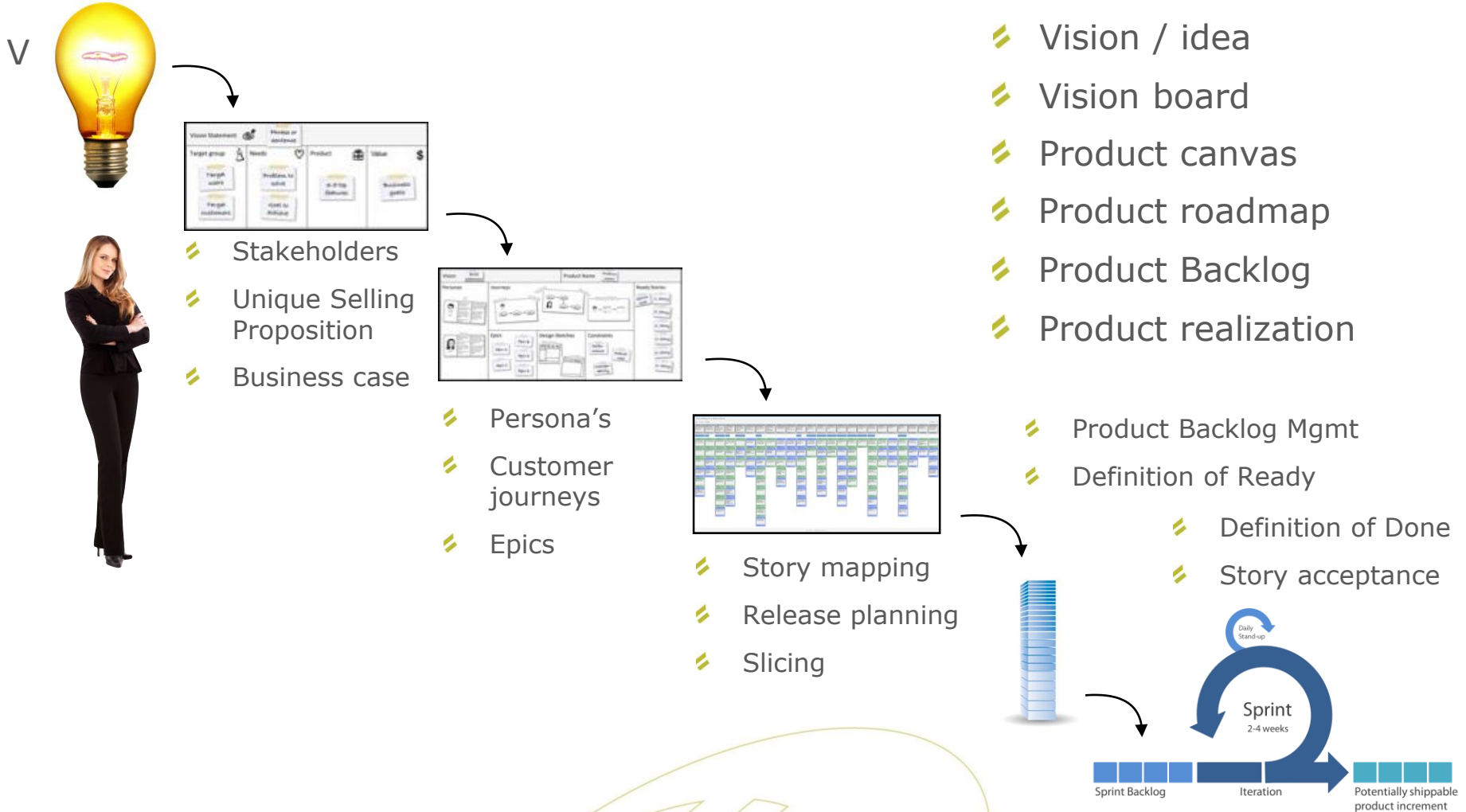




Coaching the Product Owner



Long term - from vision to product

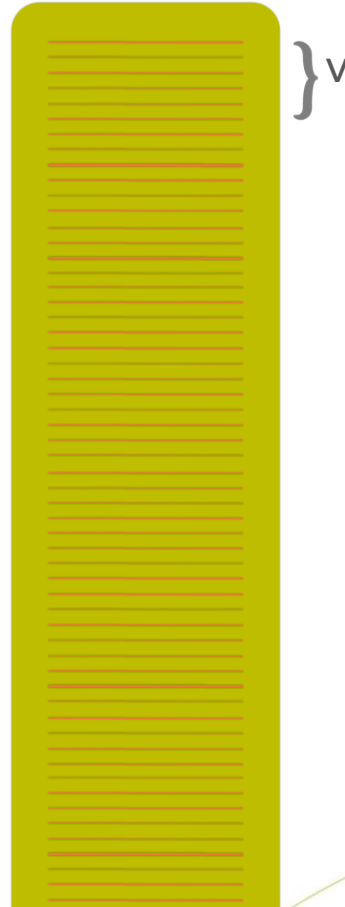


Short term - steering with a balanced backlog

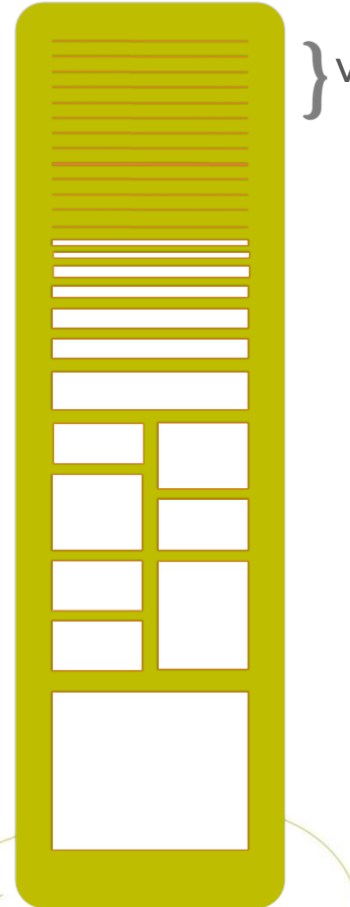
Items too big?



Too many items?



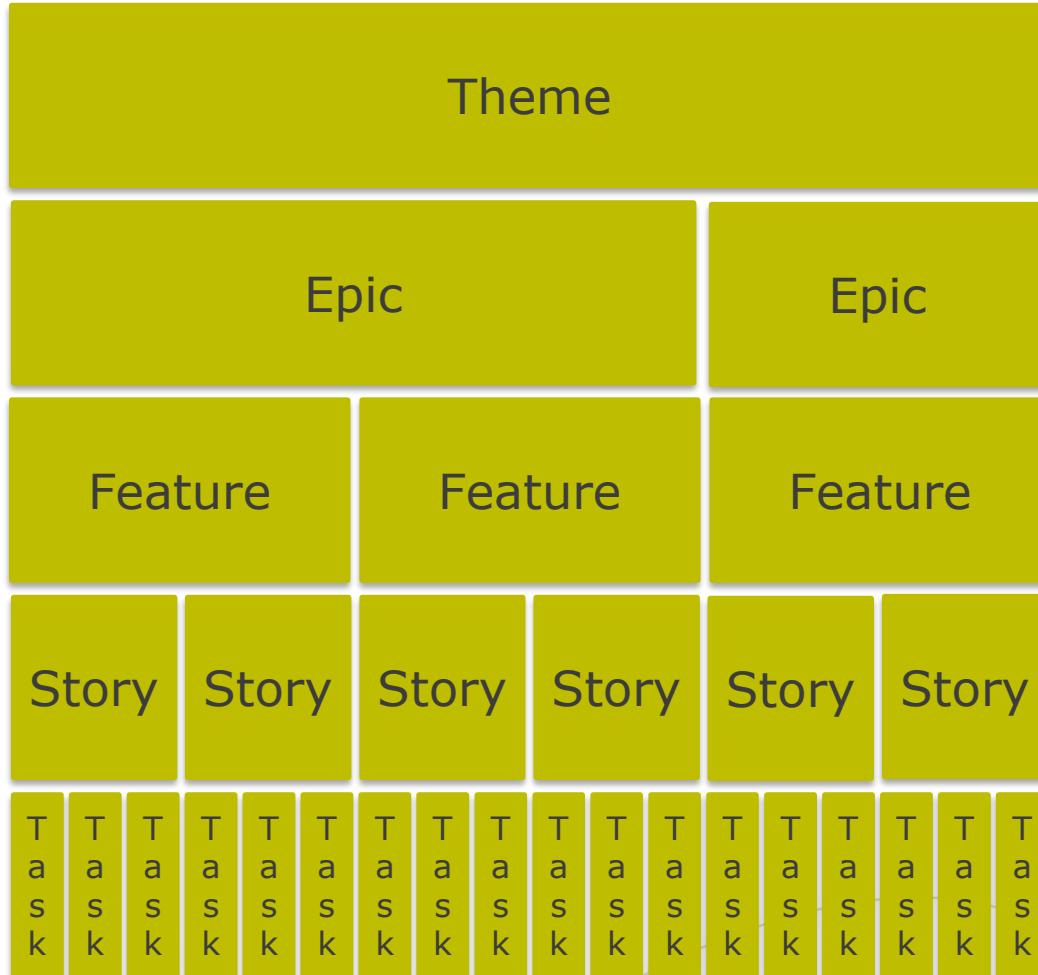
Balanced



D E E P

- ▤ Detailed sufficiently
- ▤ Estimated
- ▤ Emergent / evolving
- ▤ Prioritized

DEEP - Detailed sufficiently



⇒ Budget / capacity allocation decision

⇒ Epics span releases

⇒ Features fit in releases

⇒ Stories fit in sprint

⇒ Stories are broken down into tasks

Epic, feature and user story format

- ⚡ As a [user role]
- ⚡ I want to [result]
- ⚡ [so that [reason]]

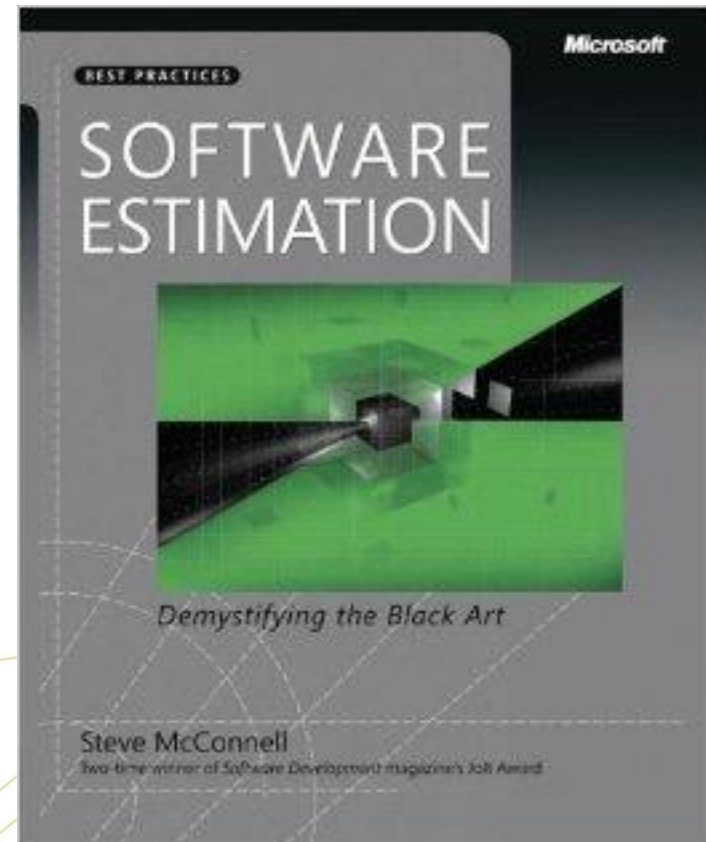
- ⚡ As a traveler
- ⚡ I want to book a hotel room online
- ⚡ so that I know where to stay the night

Spikes

- Used to analyze or answer a question
 - Yes, we continue with the story
 - No, save analysis for the future
- Training activities needed to support a story
- Time boxed in hours
- Estimated in story points



- ⚡ Estimating
 - ⚡ The art of guessing what the amount of time will be for a certain activity
- ⚡ Why do we have so much trouble with estimating?
 - ⚡ "Prediction is very difficult, especially about the future." - Niels Bohr
 - ⚡ Humans are not well suited to estimate abstract items



How to increase the accuracy of estimates?

- ⚡ Use the law of large numbers
 - ⚡ Scientific research (wisdom of the crowds) have shown that the average of group estimates is way more precise than single estimates

- ⚡ Estimate relatively instead of absolute
 - ⚡ Humans are prone to large errors when estimating absolute
 - ⚡ Humans are very well equipped to estimate on a relative scale

- ⚡ Estimate based on bandwidths instead of exact numbers
 - ⚡ E.g. with 50% certainty we will finish between 11 en 13 days, or with 90% certainty we will finish between 8 and 15 days

- ⚡ Estimate smaller items instead of large items
 - ⚡ The larger the items / task, the more uncertain the outcome will be

Estimating

- ⚡ A tool is **planning poker**



Planning poker helps because...

- ✦ Use the law of large numbers
 - ✦ We estimate items with a full group based on individual estimates
 - ✦ Be sure we don't influence each other by showing cards too soon

- ✦ Estimate relatively instead of absolute
 - ✦ Points are not absolute values, they stimulate thinking relatively
 - ✦ An 8 takes 4 times as much time as a 2

- ✦ Estimate based on bandwidths instead of exact numbers
 - ✦ The poker cards are based on bandwidths... no numbers between 8 and 13

- ✦ Estimate smaller items instead of large items
 - ✦ The lower the points, the more fine-grained you can estimate

Planning poker

- ⚡ Prepare:
 - ⚡ A prioritized list of backlog items (requirements).
 - ⚡ An analyst or business-person to elaborate on the items.
- ⚡ Action per backlog item:
 - ⚡ Each person in the group shows his/her estimation at the same time, don't think too long about it.
 - ⚡ The highest and lowest scores elaborate on their choice (short!)
 - ⚡ If needed, a short discussion about the values follows
 - ⚡ Repeat the steps above until consensus is reached

Don't give ANY information about your estimate
before showing the card!

Animal Planning poker

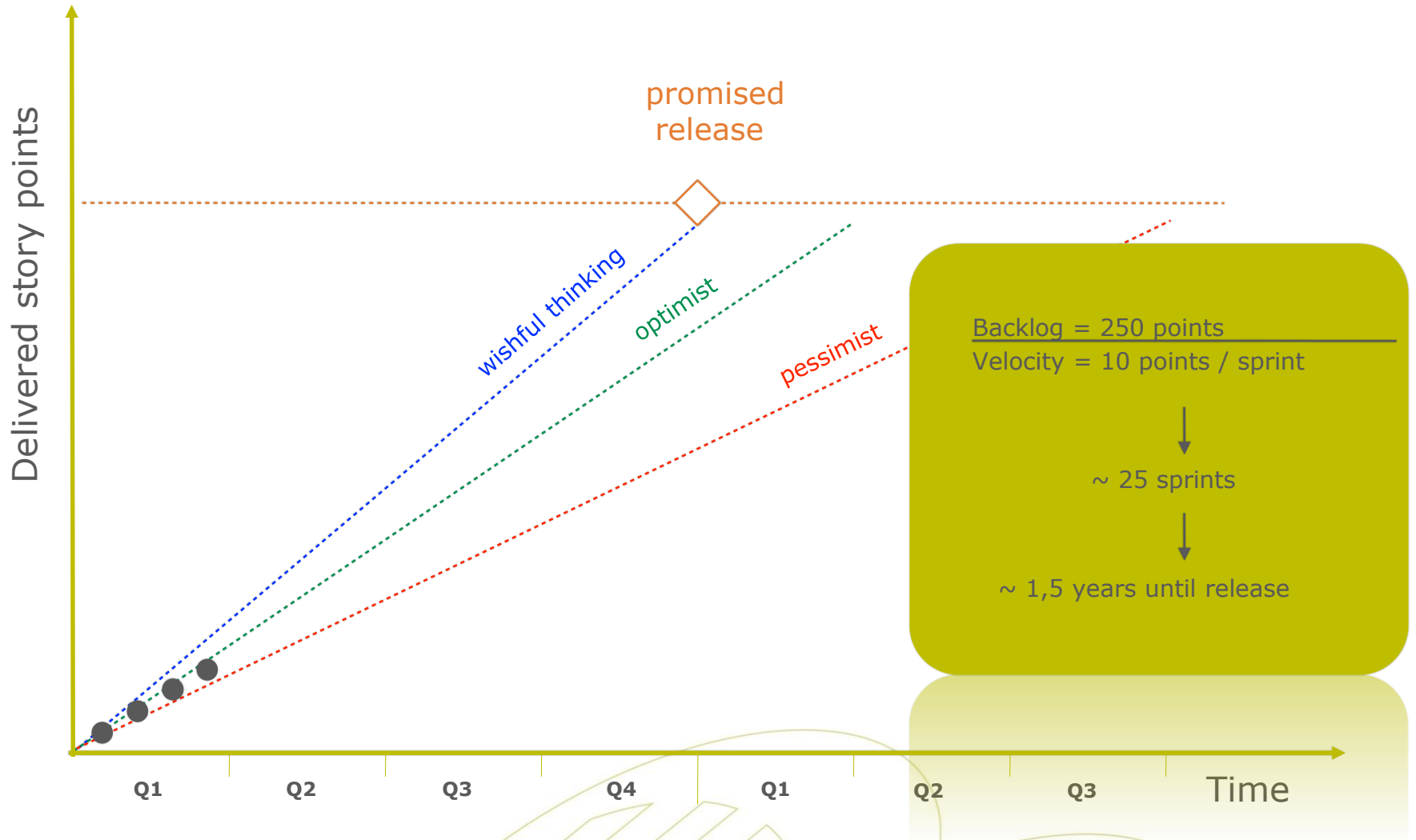
15 min



You work for a zoo, and need to buy boxes to carry the animals. The prices of the boxes depend on the height of the animal. Use planning poker to estimate the relative prices of each box

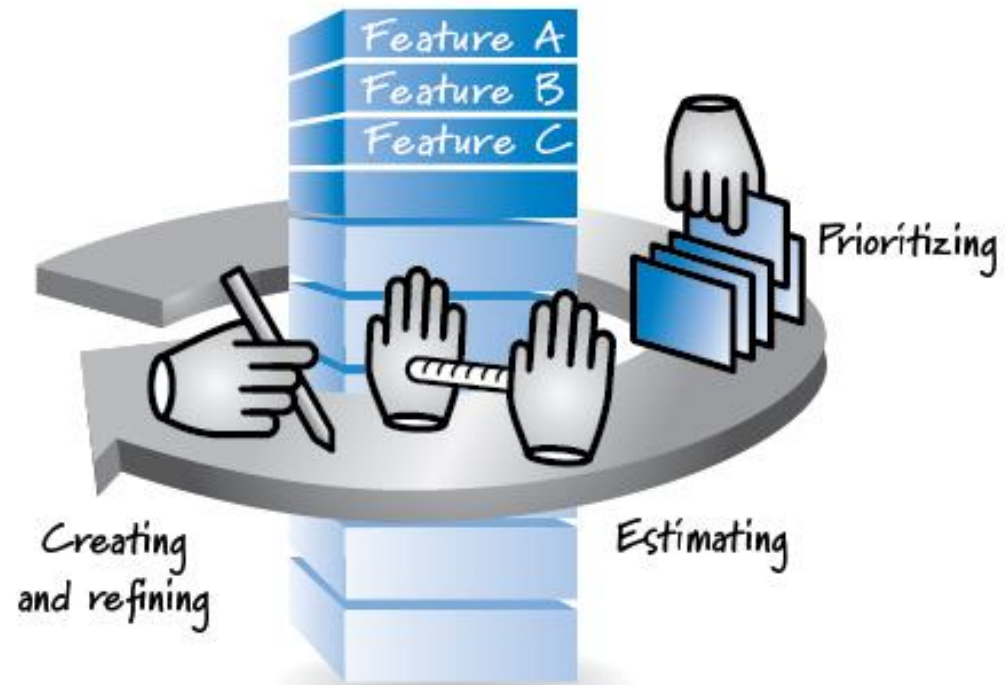
- ⚡ Dog
- ⚡ Cat
- ⚡ Elephant
- ⚡ Buffalo
- ⚡ Meerkat
- ⚡ Spider
- ⚡ Mouse
- ⚡ Rhinoceros

Release prediction



D E E P - Emergent / evolving

- Product Backlog refinement
- Slicing - principles
- INVEST - principle

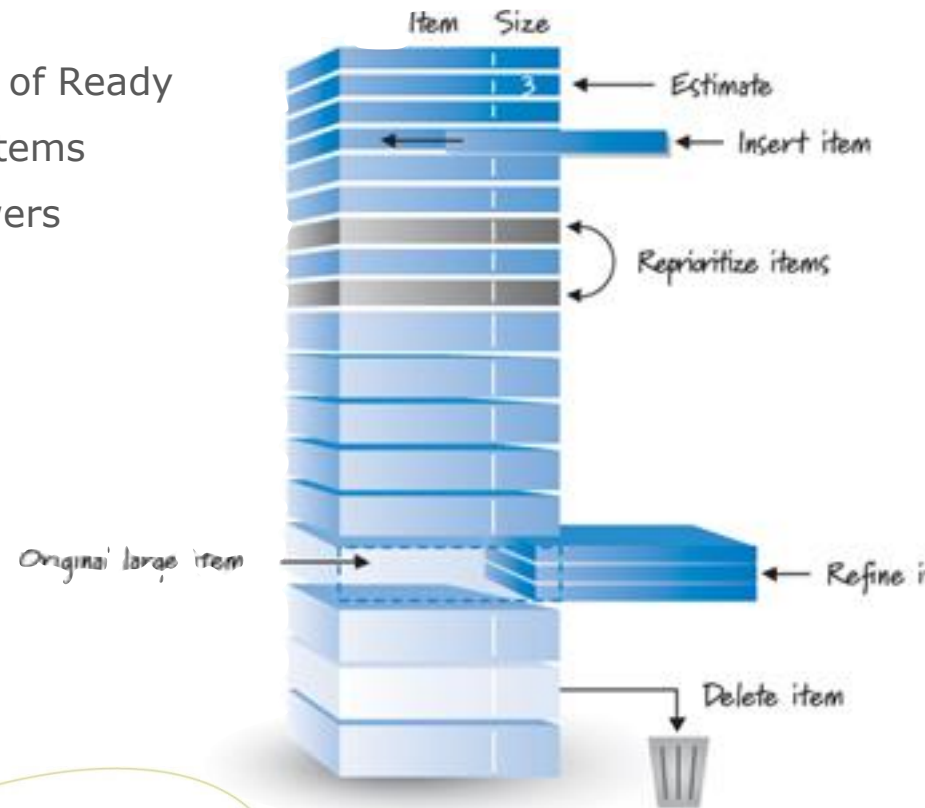


Product Backlog refinement

✦ A best practice meeting where the Product Owner and (a part of) the Development Team gather to take a look at the Product Backlog and:

- ✦ estimate new Product Backlog items
- ✦ validate new items towards the Definition of Ready
- ✦ check the general understanding of new items
- ✦ reprioritize items based on provided answers
- ✦ refine items (slice them)
- ✦ mark items that provide no further value

- ✦ About 1 - 2 hours per sprint week
- ✦ The meeting is to ensure that the next sprint planning meeting is efficient
- ✦ All items that are currently in progress (part of the sprint) are excluded



Slicing patterns

- ✦ Operations
 - ✦ Can you split the operations into separate stories?
- ✦ Business rule variations
 - ✦ Can you split the story so you do a subset of the rules first?
- ✦ Variations in data
 - ✦ Can you split the story to process one kind of data first?
- ✦ Interface variations
 - ✦ Can you split to story to handle one interface first?
 - ✦ Is there a simple version of the interface you could do first?
- ✦ Simple / complex
 - ✦ Could you split the story to do the simple parts first and a complexity later
- ✦ Defer performance
 - ✦ Could you split the story to just make it work first and enhance it to satisfy the non-functional requirement later
- ✦ Workflow steps
 - ✦ Can you split the story so you do the beginning and end workflow first and enhance with stories from the middle of the workflow later?

Slice vertical, not horizontal!

Automated Teller Machine (ATM) Horizontal and Vertical User Stories - Slicing the Cake

Vertical User Stories

Cash Withdrawal (90% Usage)

Bank Statement

Horizontal Stories

UI - PIN and Card Reader

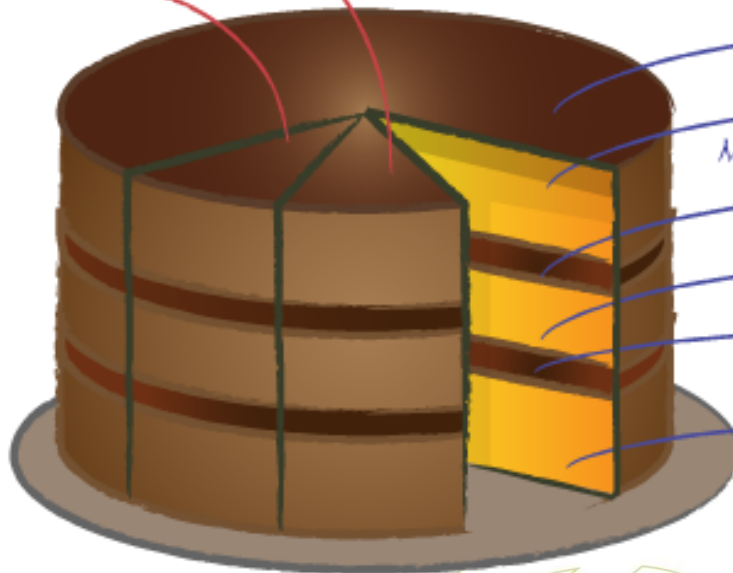
Security Layer

Middleware - Transaction Protocol

Tuxedo DB Interface

Transport Protocol

Bank Mainframe Database



Independent	The user story should be self-contained, in a way that there is no inherent dependency on another user story
Negotiable	User stories, up until they are part of an iteration, can always be changed and rewritten
Valuable	A user story must deliver value to the end user
Estimable	You must always be able to estimate the size of a user story
Small	User stories should not be so big as to become impossible to plan/task/prioritize with a certain level of certainty
Testable	The user story or its related description must provide the necessary information to make test development possible

Weighted Shortest Job First

