

ENGENHARIA DE SOFTWARE

41492-ES

Nuno Sá Couto / Rafael Direito

(nuno.sacouto@ua.pt / rafael.neves.direito@ua.pt)

Department of Electronics, Telecommunications and Informatics (DETI)

UNIVERSITY OF AVEIRO (UA), PORTUGAL

2023-24



Week #1-2

SOFTWARE DEVELOPMENT METHODOLOGIES

Software Architecture

Agenda

- + Software Development Methodologies:
 - + Agile
 - + Organizational framework
 - + Mindset, values & principles
 - + Scrum
 - + Process / Events / Artifacts
 - + Practices

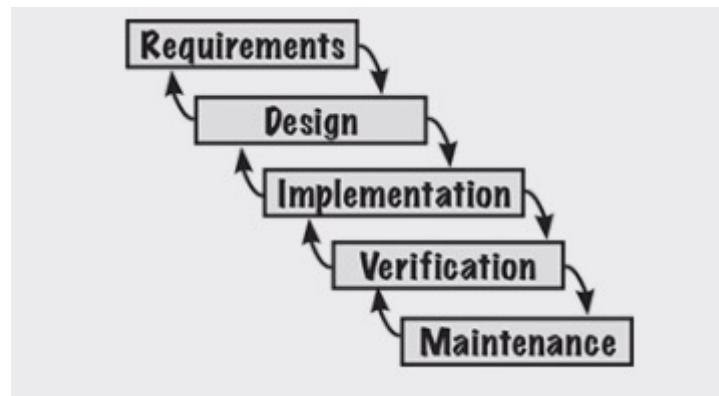
A Better Way Of Building Products



Software Development Methodologies

Waterfall Model

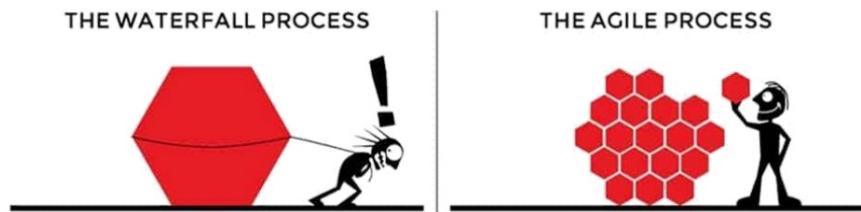
- + “Waterfall” is the name given to a specific way that software companies have traditionally built software
- + They divide their projects into phases, usually drawn in a diagram that looks like this



Software Development Methodologies

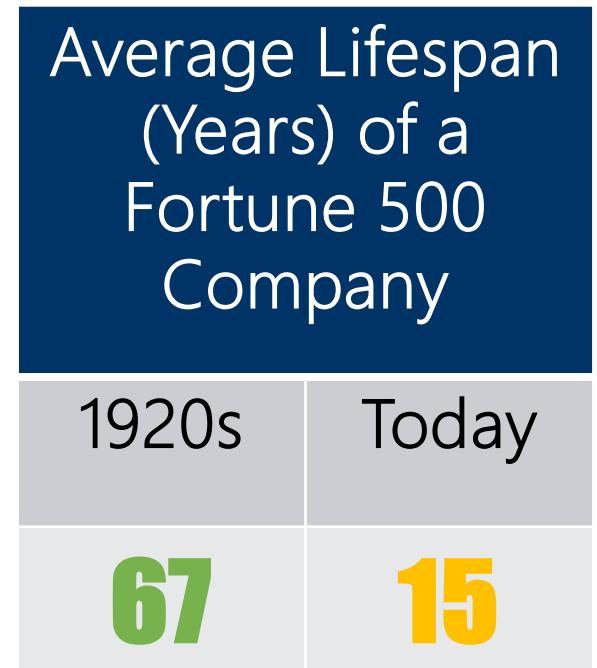
Waterfall Model – Good or Bad?

- + Waterfall isn't "good" or "bad," it's just a certain way of doing things.
- + Like any tool, it has its strengths and weaknesses.
- + However, many teams find a lot more success with agile methodologies.
- + Agile teams use their sprints and planning practices to start building the software quickly, which lets them get working software into the hands of their users



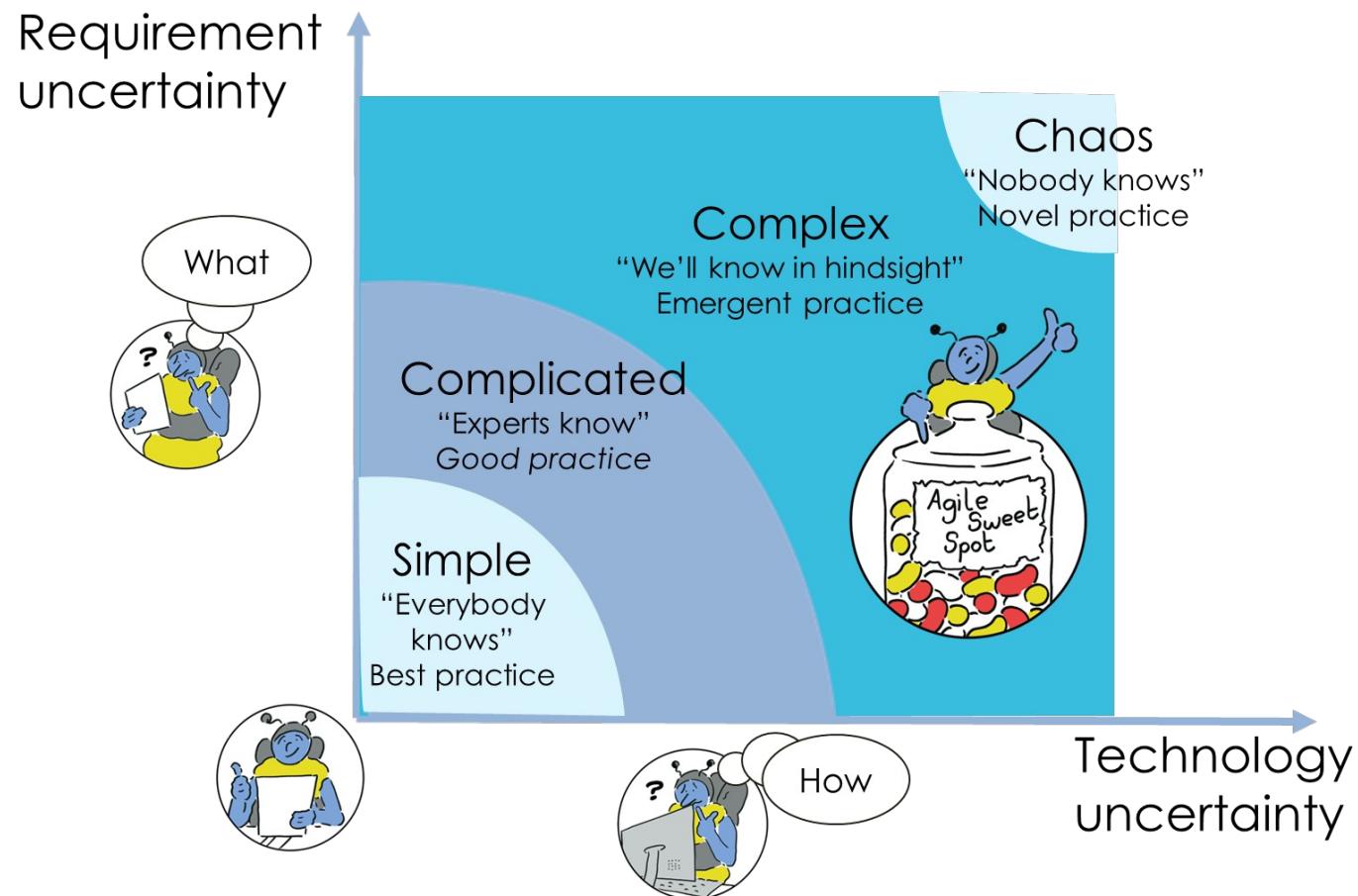
Software Development Methodologies

Providing context: Time Has Contracted for Businesses



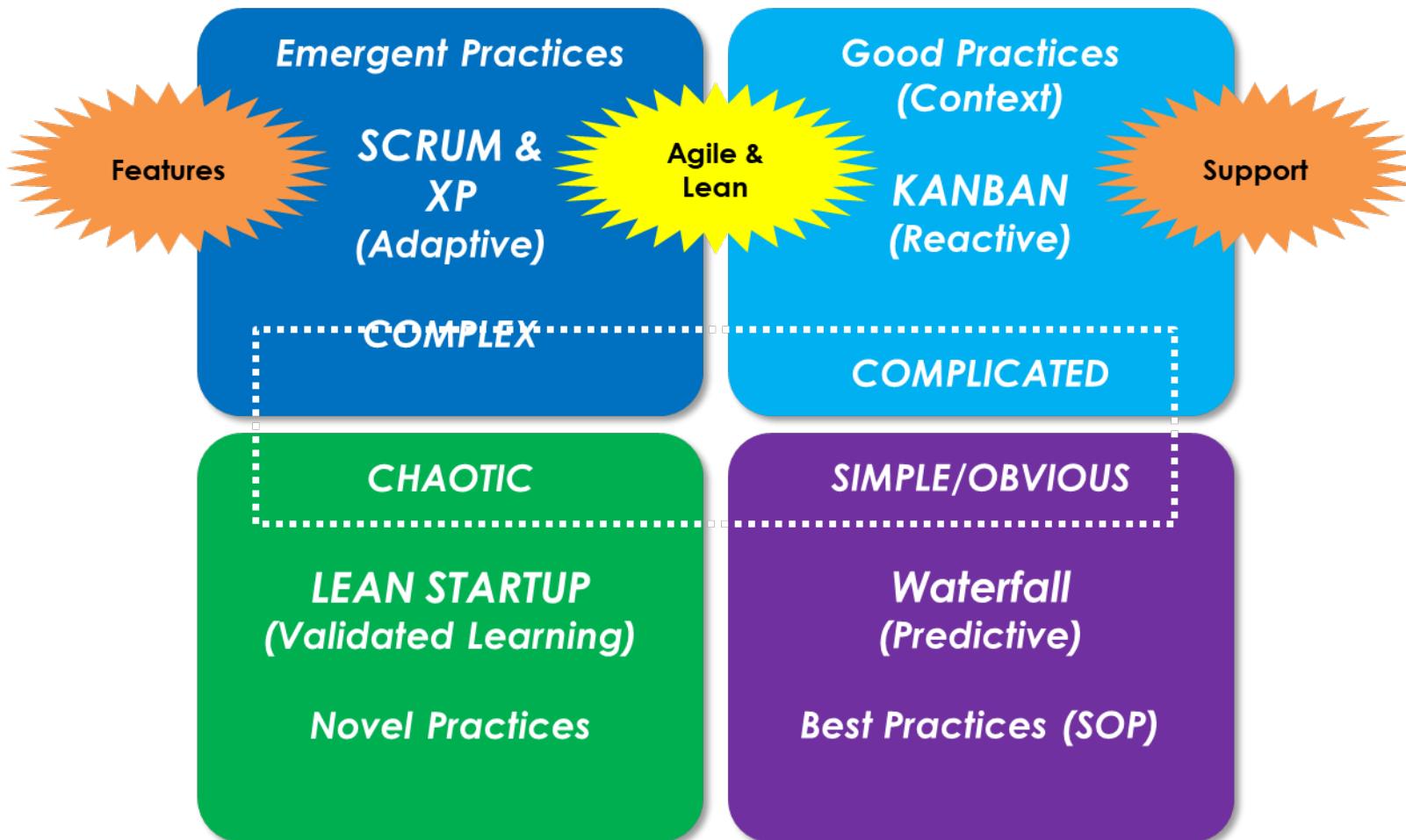
Software Development Methodologies

Providing context: Stacey Complexity Model



Software Development Methodologies

Providing context: Cynefin Decision-Making Framework



Software Development Methodologies

Providing context: Reasons

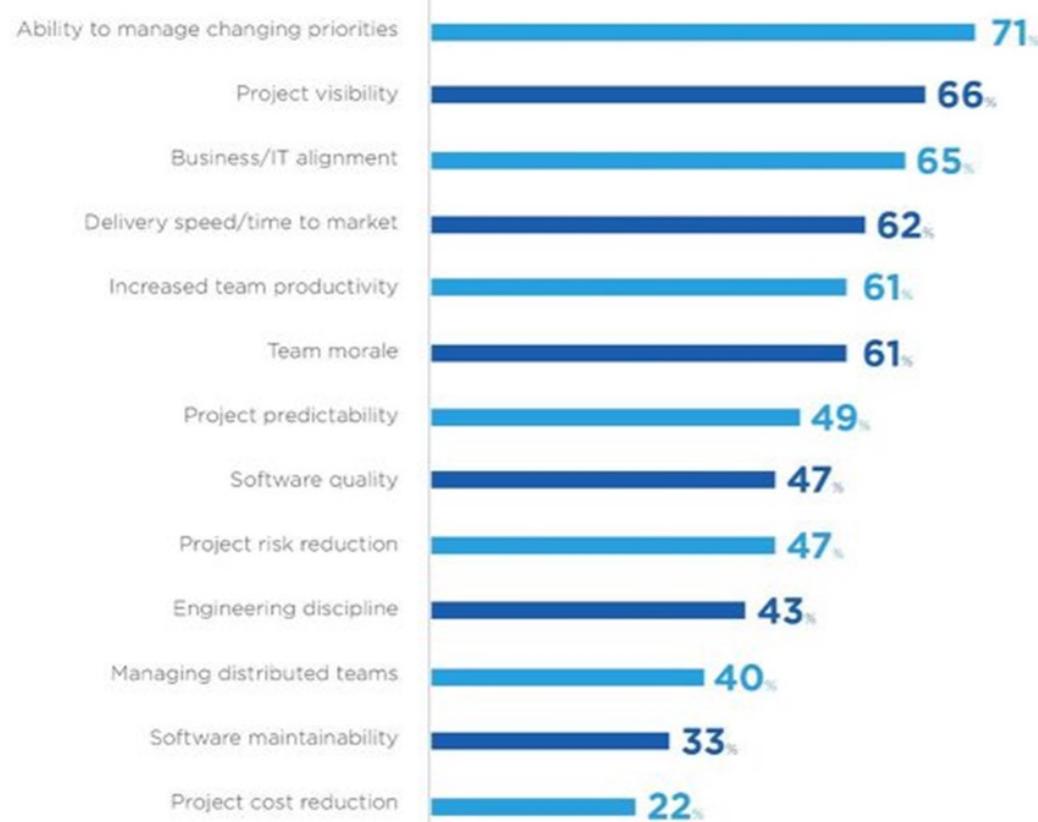
Reasons for Adopting Agile



*Respondents were able to make multiple selections.

Benefits of Adopting Agile

By implementing agile, respondents cited seeing improvements in the following areas:



The Agile Mindset, A Different Way of Working

A PHONE CALL WITH

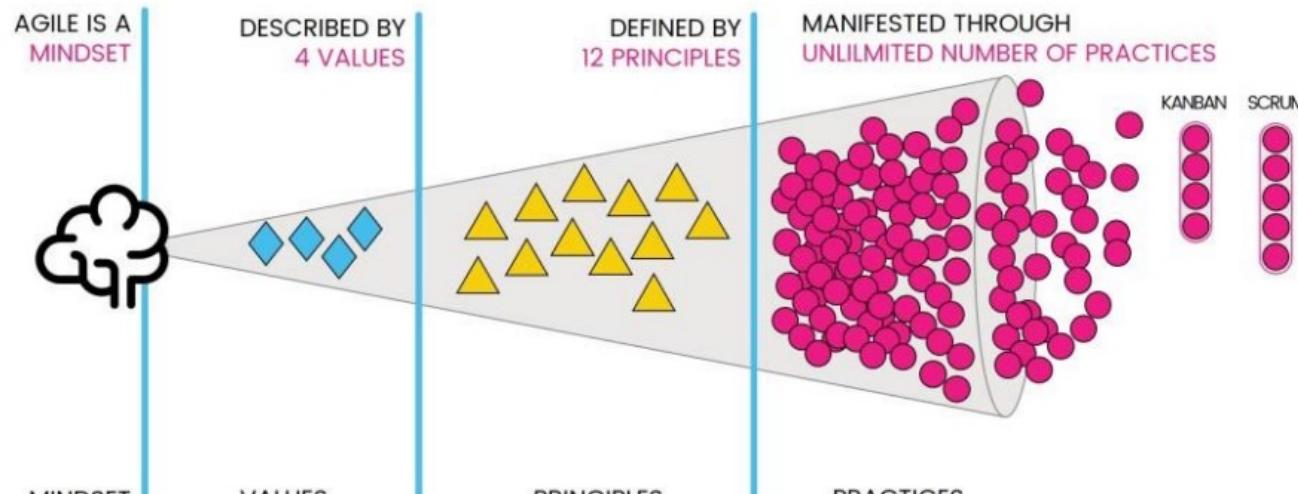
Ghmed Sidky

THE AGILE MINDSET

Software Development Methodologies

All about

MINDSET



Adapted from Ahmed Sidky's Agile Mindset

BEING AGILE

200% of the Benefit

- Customer Satisfaction
- Continuous Learning
- Leadership at all levels
- Cross Org Engagement

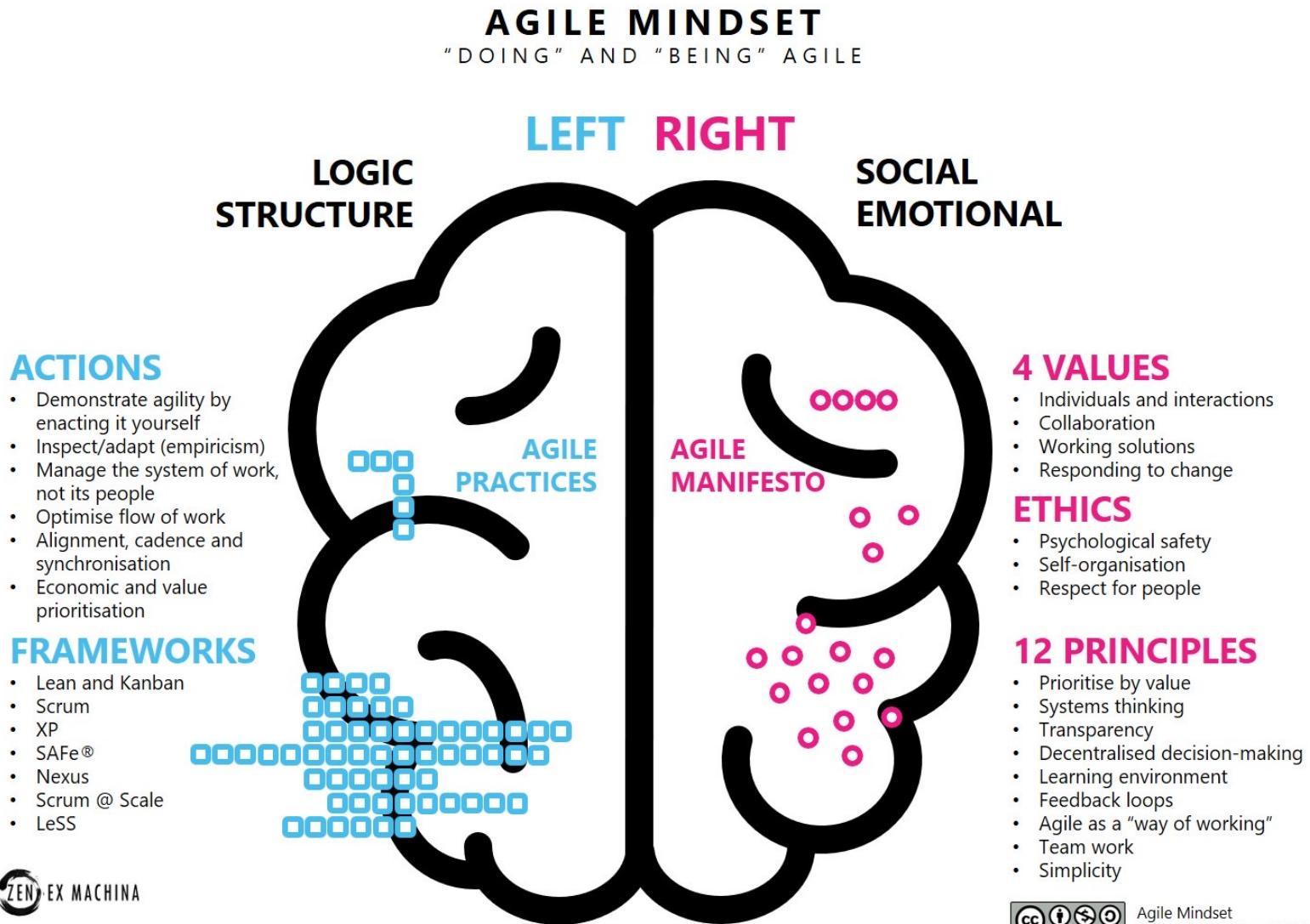
DOING AGILE

20% of the Benefit

- Greater work visibility
- Higher productivity
- Handle priority changes

Software Development Methodologies

Agile Mindset



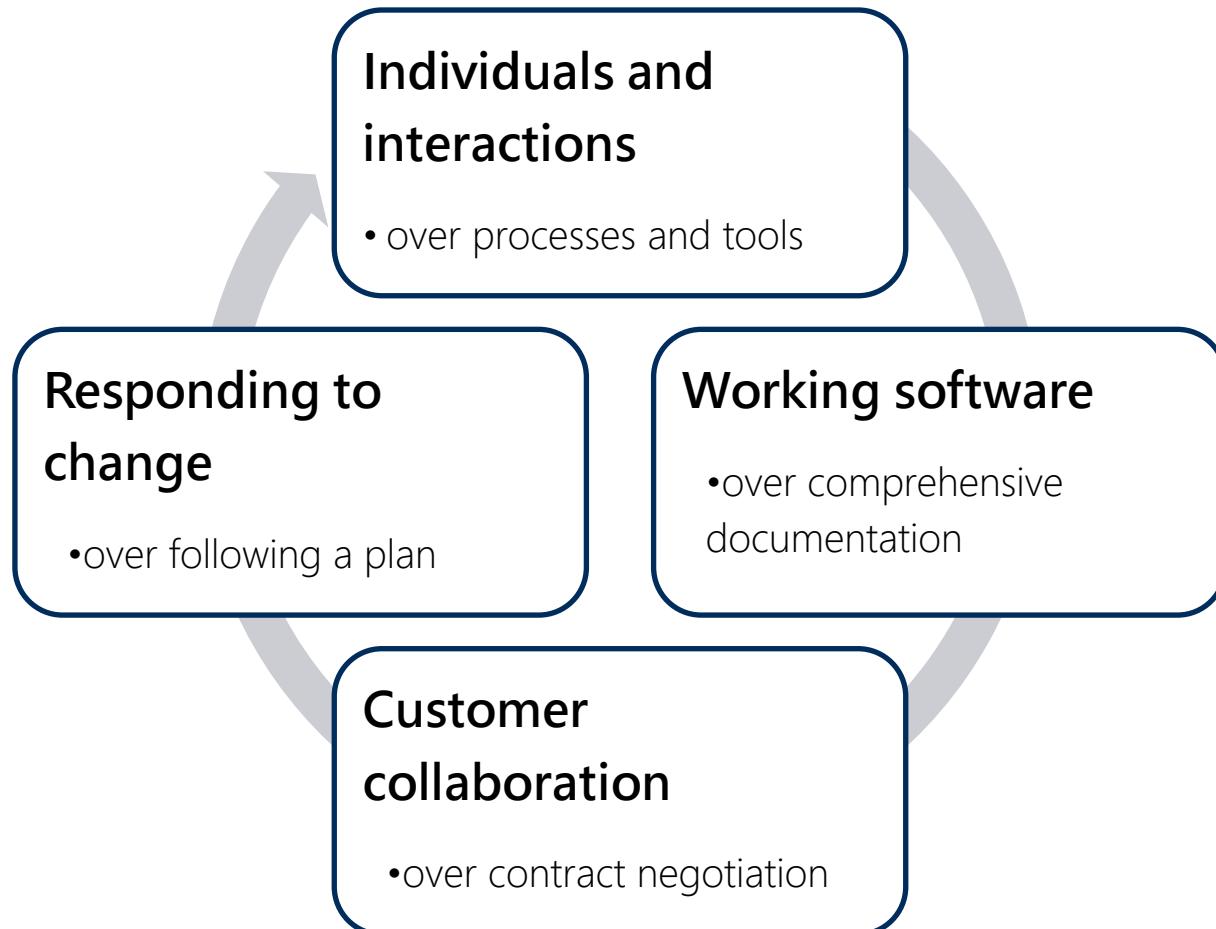
Software Development Methodologies

Agile Mindset – tick your mind-set

	Fixed Mindset	Growth (Agile) Mindset
Perspective	“Are you not smart enough to solve it”; “Right now”	“.... or have you just not solved it yet”; “Power of Yet”
Intelligence & Personality	Is Static	Can Be Developed
Leads to a Desire	To “Look” Smart	To Learn
Challenges	Tendency to Avoid	Tendency to Embrace
Obstacles & Deficiencies	Gives Up Easily Hide	Persists in the Face of Setbacks; Overcome (Perseverance & Resilience)
Effort	Fruitless or Worse	The Path to Mastery
Criticism	Ignores Useful Negative Feedback	Learns from Criticism
Success of Others	Feels Threatened	Finds Lessons & Inspiration
Result	May Plateau Early & Achieve Less Than Full Potential	Reach Ever-Higher Levels of Achievement
Viewpoint	Deterministic	Free Will

Software Development Methodologies

Agile Manifesto – The Values



Software Development Methodologies

Responding to change vs following a plan



Software Development Methodologies

Agile Manifesto – The Principles

01

Our highest priority is to satisfy the customer



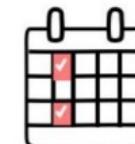
02

Welcome changing requirements, even late in development.



03

Deliver working software frequently



04

Business people and developers must work together



05

Build projects around motivated individuals: Environment, support and trust



06

Face-to-face, personal and direct team communications



Software Development Methodologies

Agile Manifesto – The Principles

07

Working software is the primary measure of progress



10

Simplicity--the art of maximizing the amount of work not done

11

The best architectures, requirements, and designs emerge from self-organizing teams.

12

08

Agile processes promote sustainable development



09

Continuous attention to technical excellence and good design enhances agility.



At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Software Development Methodologies

Agile Manifesto – Paradigm shift

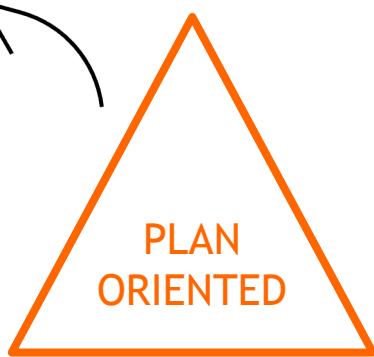
PREDICTIVE PROCESSES

ADAPTATIVE PROCESSES

RESTRICTIONS

Plan
Estimates
Time &
Cost

SCOPE



TIME

COST

VALUE
ORIENTED

Product vision
creates
functionality
estimates

TIME

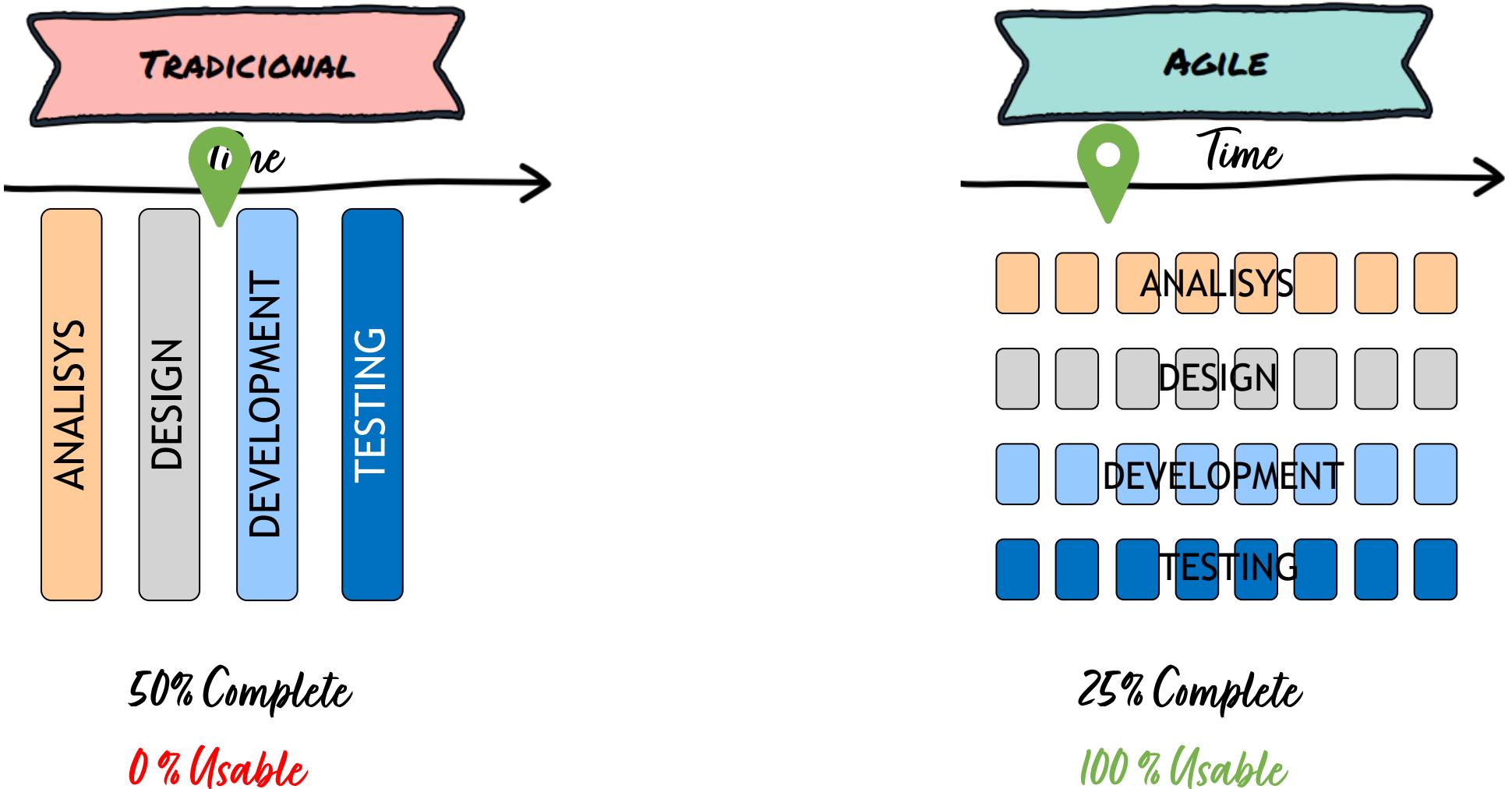
COST

SCOPE

ESTIMATIONS

Software Development Methodologies

Agile Manifesto – Paradigm shift (2)



Software Development Methodologies

Agile Manifesto – Paradigm shift (3)

TRADICIONAL

Plan everything at the begining

Fixed Scope

Deliverable based

Scope must be closed **as soon** as possible

Avoid Changes

Formal relationship with Customer

Teams **without** autonomy

Best for projects with **LOW**
degree of uncertainty

AGILE

Incremental planning & execution

Fixed Time & Cost per increment

Focus on features

Scope must be closed **as late** as possible

Changes are welcome

Foster collaboration with Customer

Teams **with** autonomy

Best for complex projects

Software Development Methodologies

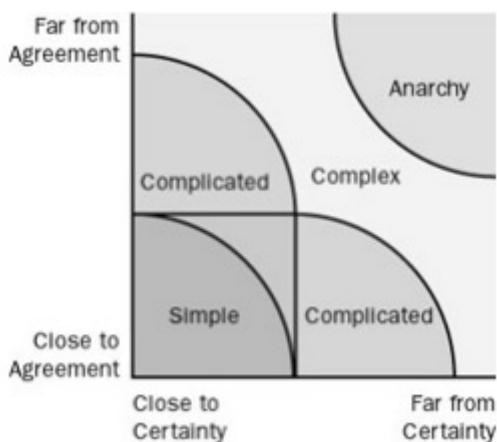
Agile Manifesto – Paradigm shift (4)



Agile is an **empirical well-defined** control process for complex projects

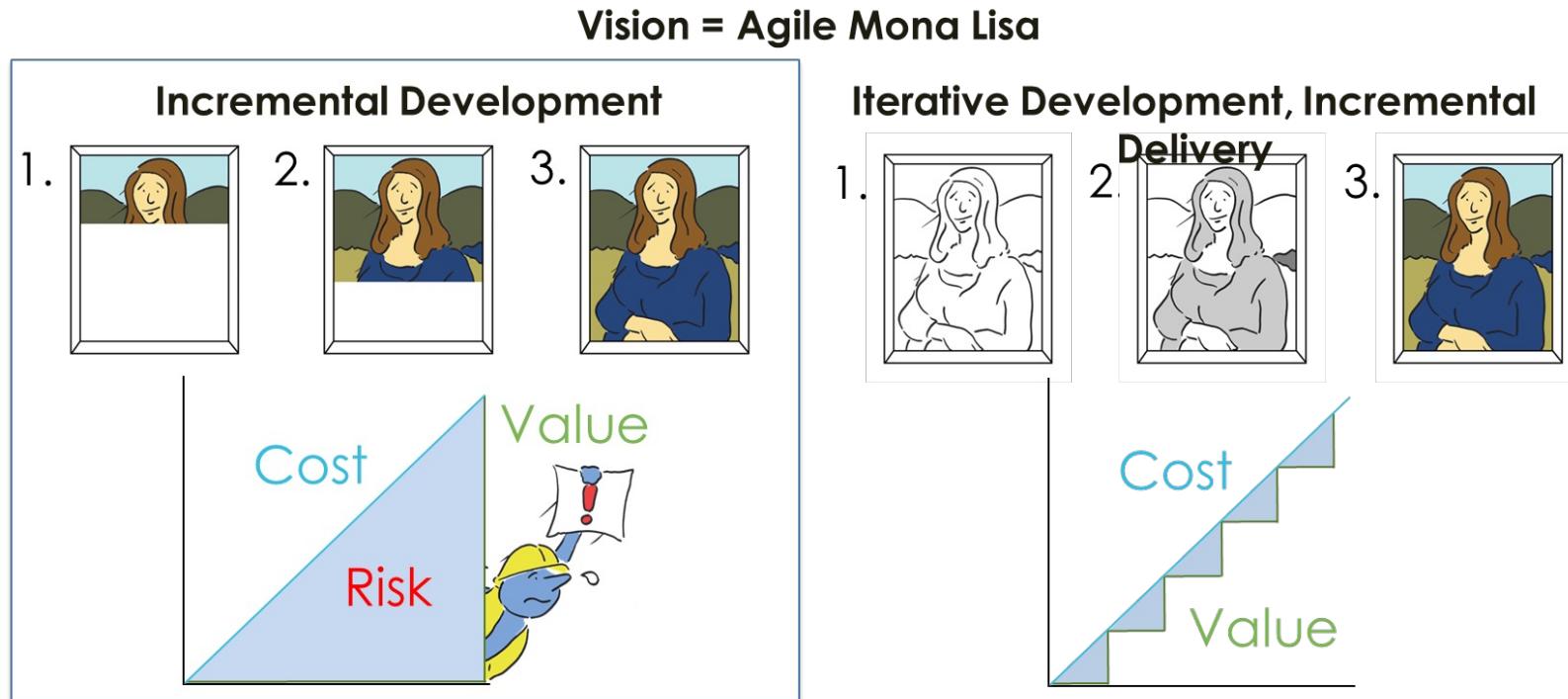


decisions are made based on **observation and experimentation** rather than on **detailed upfront planning**



Software Development Methodologies

Incremental vs Iterative

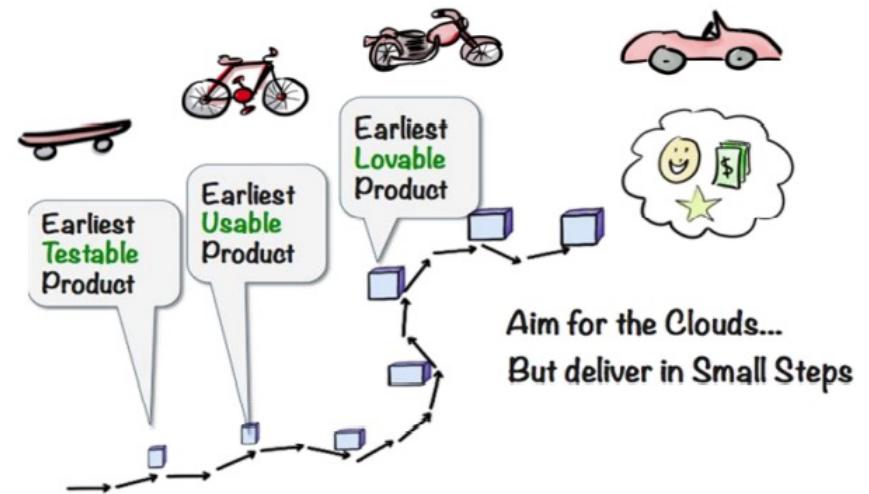
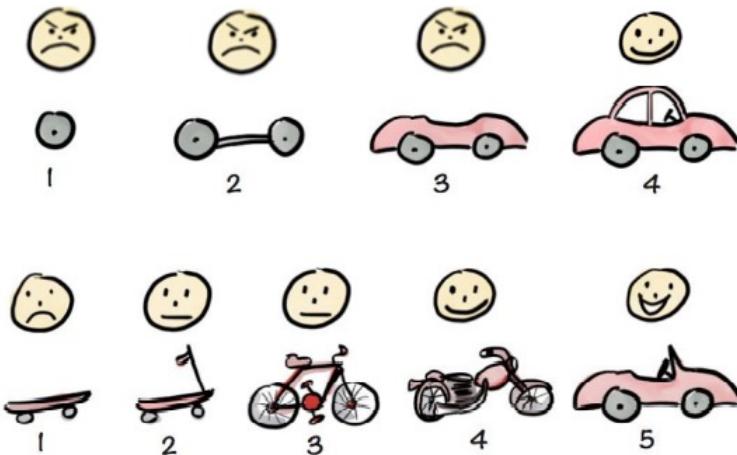


Benefits of Iterative Development with Incremental Delivery

- Faster time to market
- Responsive to changing priorities
- Higher productivity
- Higher quality
- Reduced risk
- Better customer satisfaction
- Better employee engagement
- Better alignment of IT and business

Software Development Methodologies

Agile Manifesto – Agile Concepts: Minimum Viable Product (MVP)



Earliest Testable Product is the skateboard or bus ticket – the first release that customers can actually do something with.

Earliest Usable Product is perhaps the bicycle. The first release that early adopters will actually use, willingly.

Earliest Lovable Product is perhaps the motorcycle. The first release that customers will love, tell their friends about, and be willing to pay for.

Earliest Feedbackable Product - basically the paper prototype or equivalent that you use to get your first feedback from the customer

Software Development Methodologies

Agile Manifesto – Empirical Control Process – 3 core concepts

VISIBILITY

- Aspects that affect the outcome must be visible
- **MUST** be also **true**. No room for deceiving
- **Done** vs **Not Done**

INSPECTION

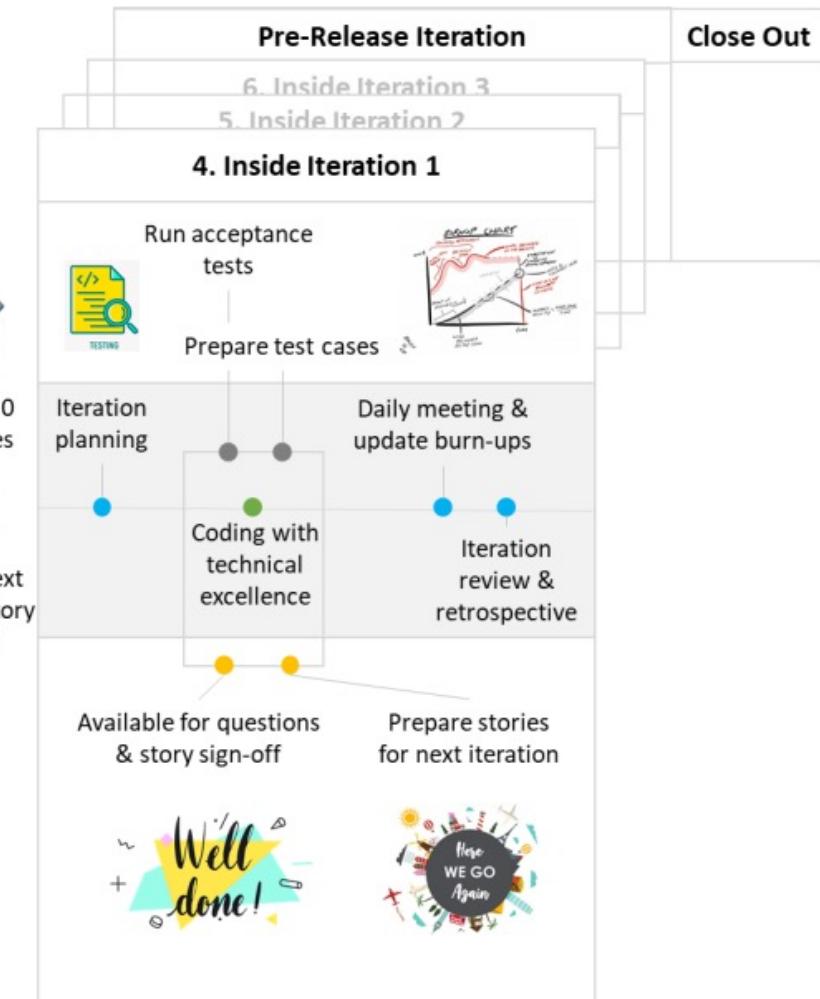
- Aspects of the process must be inspected frequently
- Sets the Working Cadence
- Inspector must have skills to make assessment

ADAPTATION

- Aspects of the process or/and resulting product will not be acceptable > adjust process or material being produced
- Must be made as quickly as possible to minimize deviation

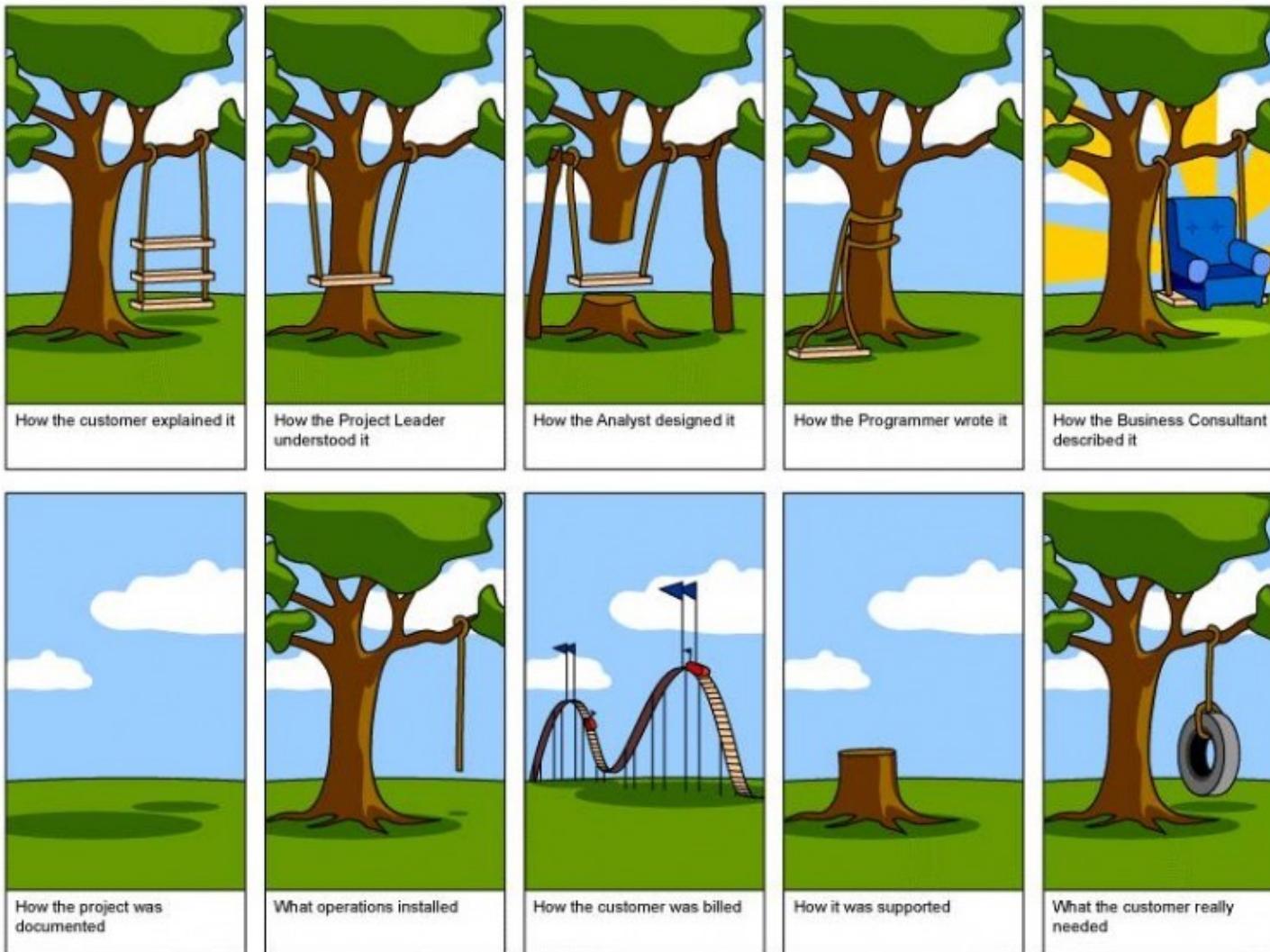
Software Development Methodologies

Agile Manifesto – Agile Process



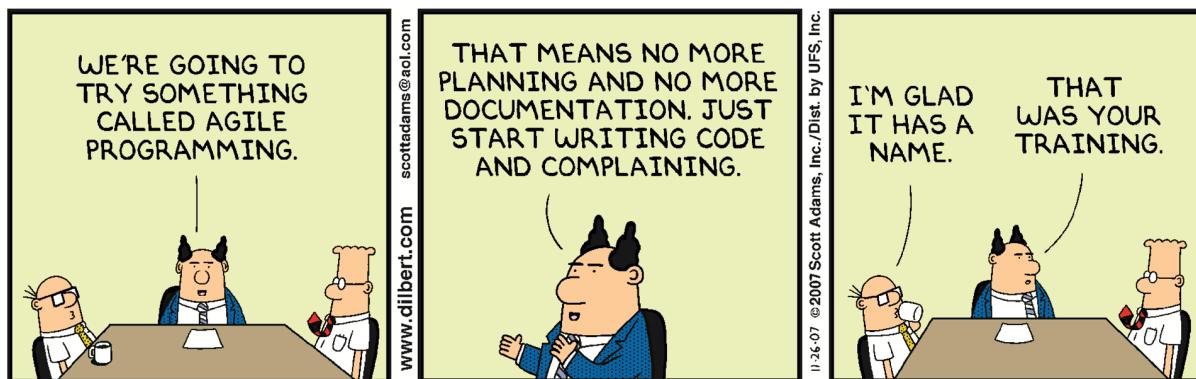
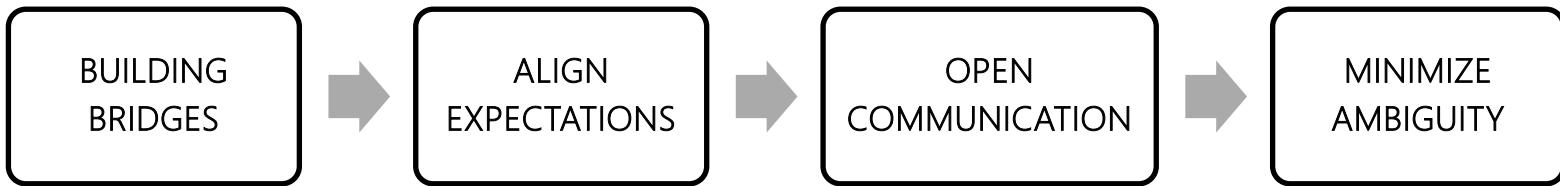
Software Development Methodologies

Traditional Projects: Building the Wrong thing



Software Development Methodologies

Agile Manifesto – Agile Benefits: Communication, Collaboration and Culture



Software Development Methodologies

Agile Manifesto – Agile Benefits: Commitment



CUSTOMER MUST BE INVOLVED AND COMMITTED



DEV TEAM MUST BE COMMITTED TO PRODUCT VISION, STABLE AND WITH MULTIFUNCTIONAL

Software Development Methodologies

Exercise – Marshmallow challenge



**4 People
18 Minutes!**

- Teams of 4
- Build the **Tallest Freestanding Structure**. Flat on the table with no additional furniture, etc. Not suspended by anything.
- The **Entire Marshmallow Must Be On Top**. Do not cut the marshmallow into pieces.
- Use As Much or As Little of the Kit. Kit includes 20 spaghetti sticks, 1 yard of tape, and 1 yard of string plus 1 marshmallow.
- Break Up Materials. The team can cut/break the spaghetti sticks, tape and string as desired.
- No other materials can be used.
- The Challenge Lasts **18 Minutes**. The structure must stand free for at least **60 seconds** without any physical assistance by the team or other materials.

Software Development Methodologies

Exercise – Marshmallow challenge – Reflections



The slide is titled "The Challenge" in large, bold, black font. Below the title is a cartoon illustration of four people sitting around a table covered with a white cloth, working together to build a structure. To the right of the illustration, the challenge details are listed:

- Eighteen Minutes**
- Teams of Four**
- Tallest Freestanding Structure**

Below the text are four items used in the challenge, each with a label:

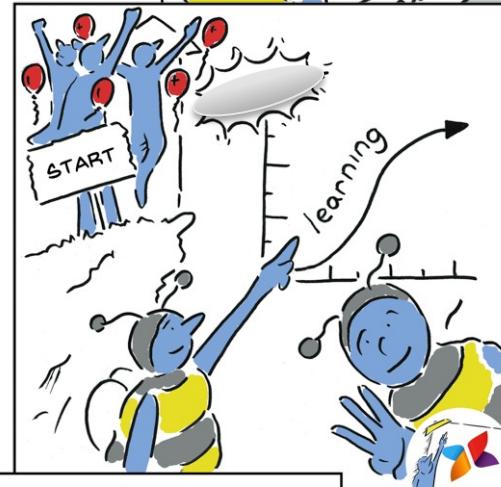
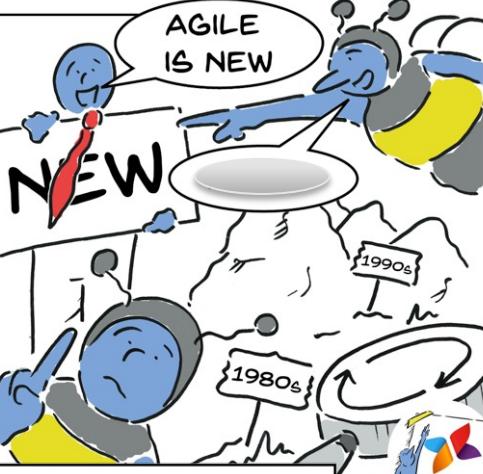
- 20 sticks of spaghetti
- + one yard tape
- + one yard string
- + one marshmallow

Software Development Methodologies

Agile Myths

MYTH 1

False



MYTH 3 AGILE GIVES
INSTANT BENEFIT

False

MYTH 4 AGILE MEANS NO DOCUMENTATION

False



MYTH 2 IMPLEMENTING AGILE IS EASY

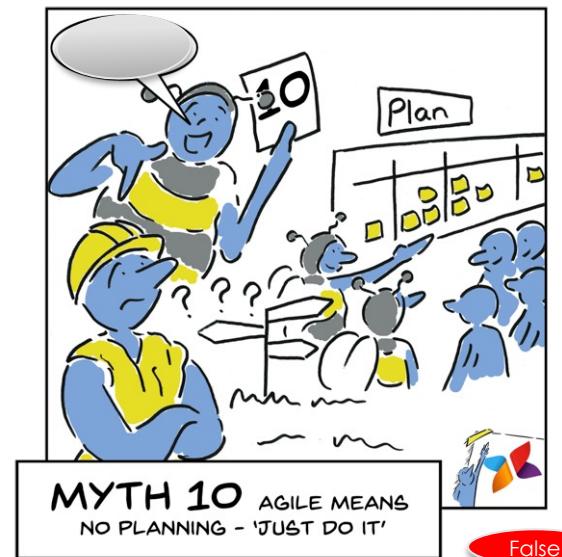
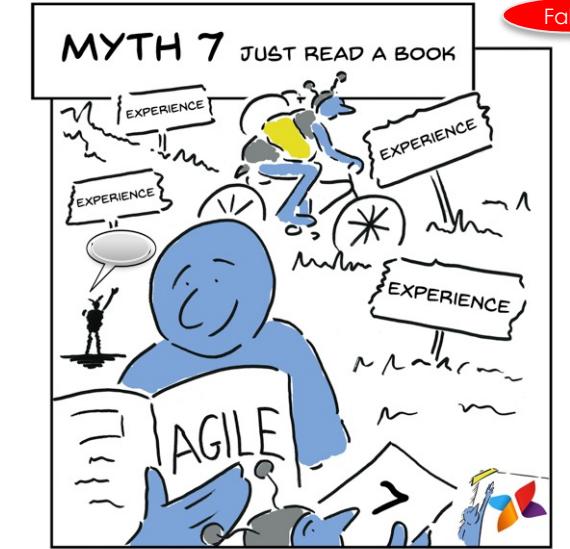
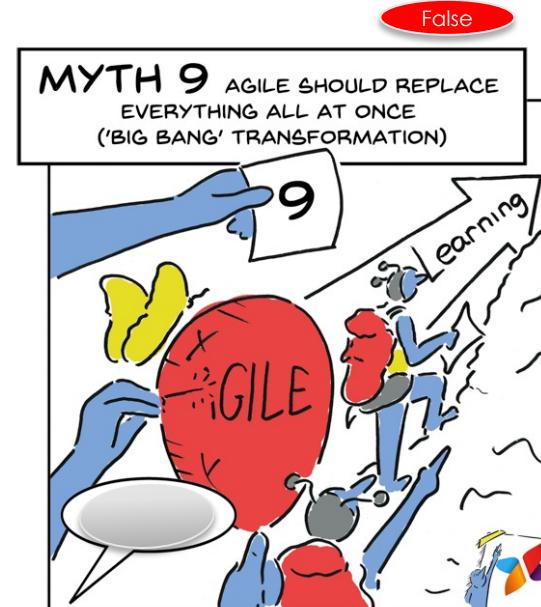
False



MYTH 5 AGILE MEANS
'HACKING' CODE TOGETHER WITH
LITTLE THOUGHT OR DESIGN

False

Software Development Methodologies



Software Architecture

Agenda

- + Software Development Methodologies:
 - + Agile
 - + Organizational framework
 - + Mindset, values & principles
 - + Scrum
 - + Process / Events / Artifacts
 - + Practices



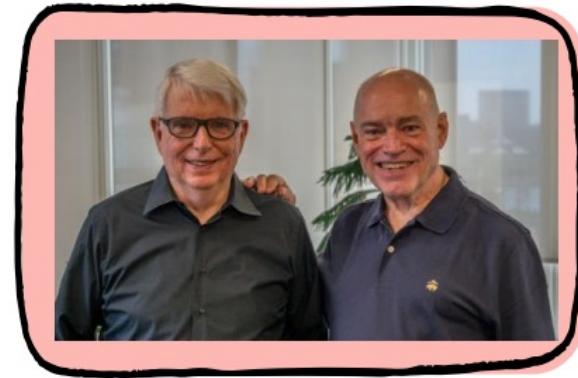
Software Development Methodologies

SCRUM – a brief history



1986

"The New Product Development Game"



1995

Scrum

2010

"The Scrum Guide"

The New New Product
Development Game (hbr.org)

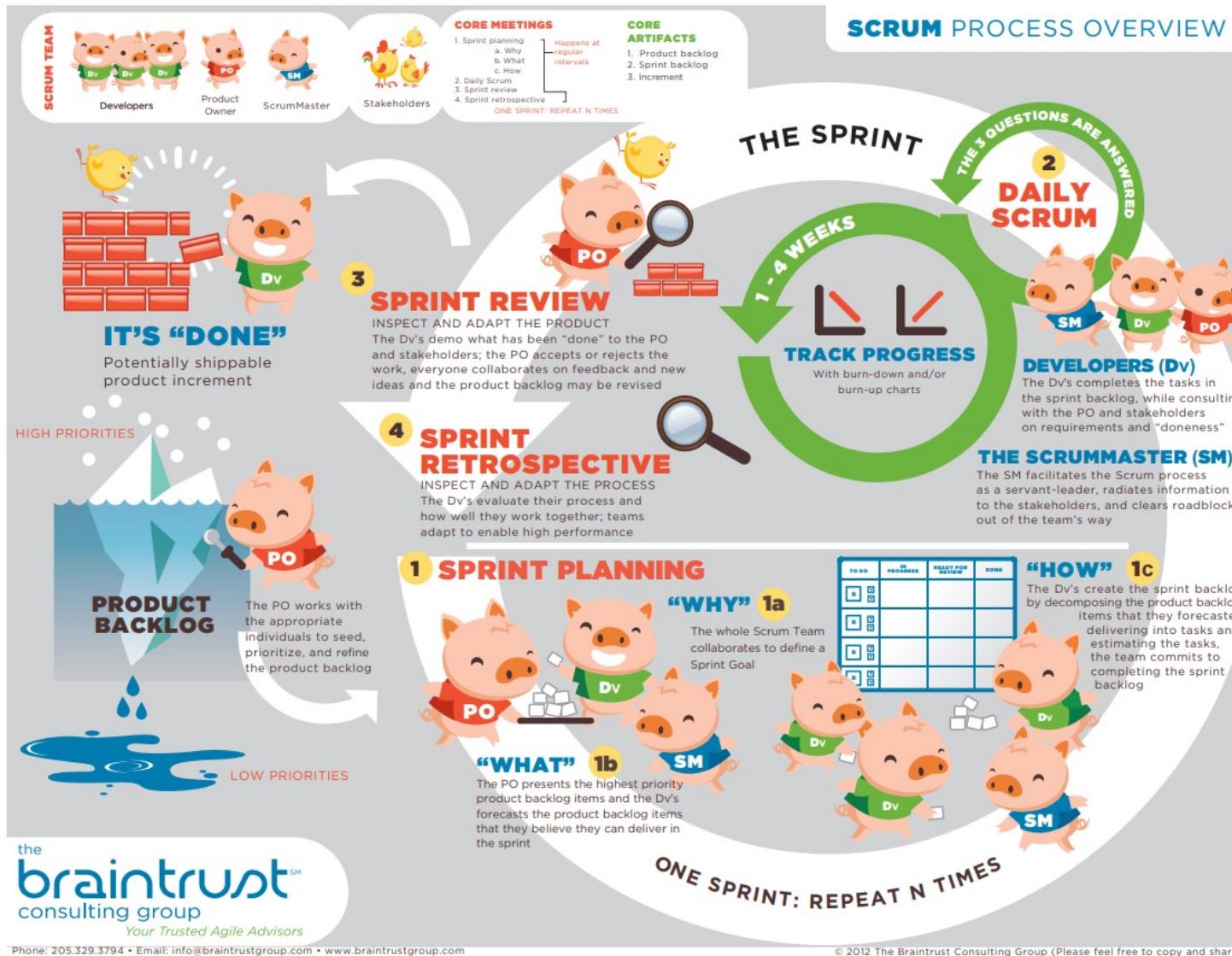
Software Development Methodologies

SCRUM – Core Values > Your values if you choose to enter in a agile team ;)



Software Development Methodologies

SCRUM – 5 Levels of Planning



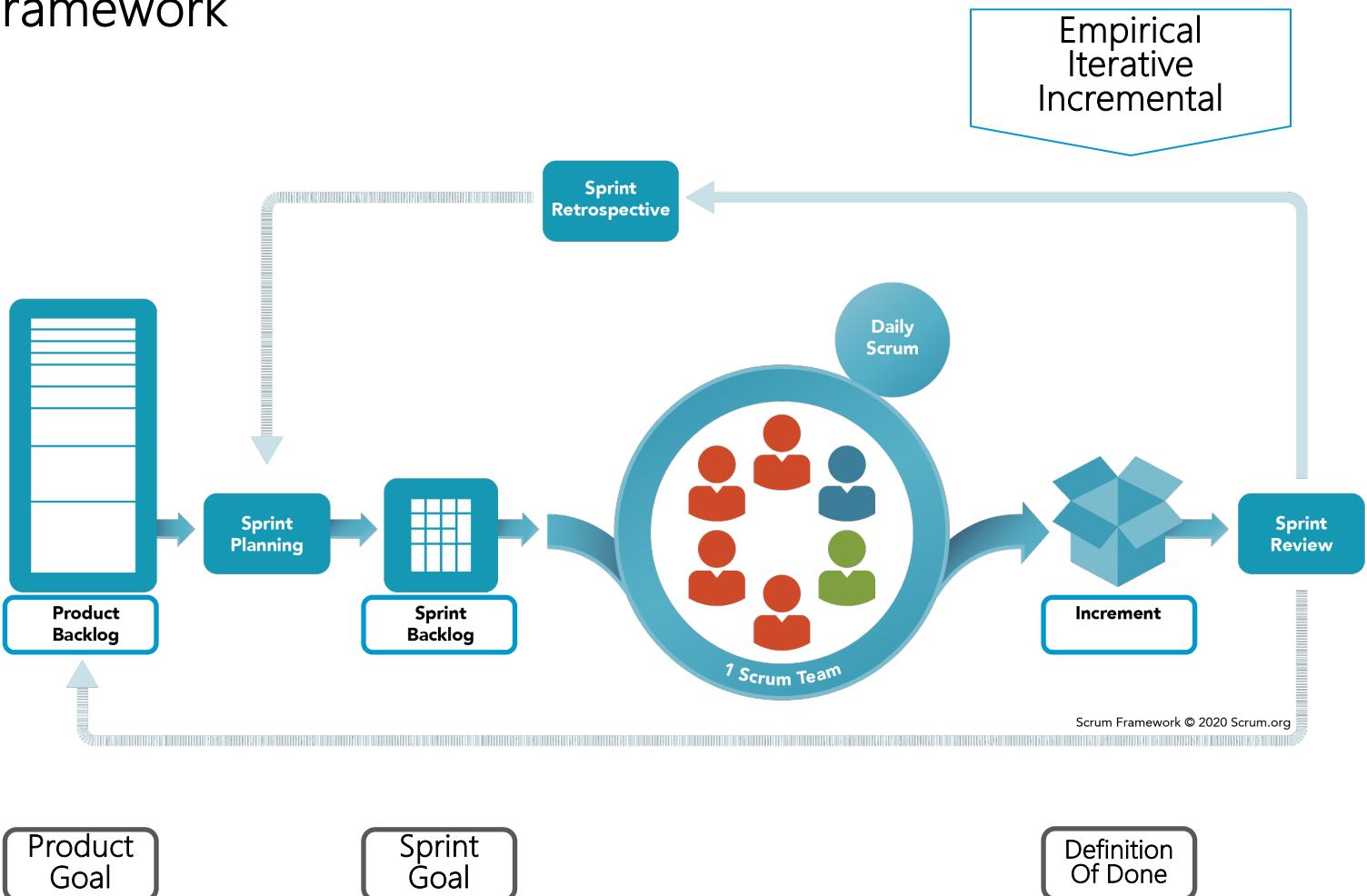
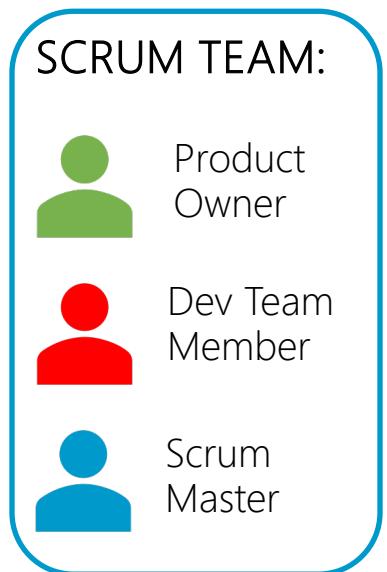
Software Development Methodologies

SCRUM – Core and Non-Core Components

	TEAM	EVENTS	ARTIFACTS
Scrum Core	<ul style="list-style-type: none">○ Developers○ Product Owner○ Scrum Master	<ul style="list-style-type: none">○ Sprint○ Sprint Planning○ Daily Scrum○ Sprint Review○ Sprint Retrospective	<ul style="list-style-type: none">○ Product Backlog○ Sprint Backlog○ Increment
Out of Scrum Core	<ul style="list-style-type: none">○ Stakeholders	<ul style="list-style-type: none">○ Visioning○ Product Roadmapping○ Release Planning○ Backlog Refinement○ Sizing	<ul style="list-style-type: none">○ Product Vision○ Product Roadmap○ Release Plan○ Burn Down / Up Charts

Software Development Methodologies

SCRUM – Process Framework



Software Development Methodologies

Scrum Team – Max. 10 pax; ideal: 7/8 pax



Developers



Product
Owner



ScrumMaster

- + Small team of people, and large enough to complete significant work within a Sprint
- + One Scrum Master, one Product Owner, and Developers
- + There are no sub-teams or hierarchies
- + Focus on one objective at a time, the Product Goal
- + Is responsible for all product-related activities
- + Is accountable for creating a valuable, useful Increment every Sprint

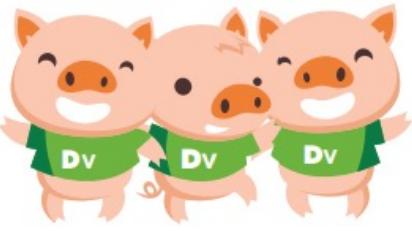


External
Scrum Team
Stakeholders

Stakeholders

Software Development Methodologies

Scrum Team – Development Team



Developers

- + People committed to create usable increments each Sprint
- + Broad skill-set and will vary with the domain of work. 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.

Software Development Methodologies

Scrum Team – Product Owner



Product
Owner

- + Accountable for maximizing the value of the product resulting from the work of the Team
- + Is one person. Represents the needs of the many stakeholders in the Product Backlog
- + Accountable for Product Backlog management:
 - + 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

Software Development Methodologies

Scrum Team – Scrum Master

- + Is accountable for establishing Scrum as defined in the Scrum Guide, in theory and practice
- + Accountable for the Scrum Team's effectiveness
 - + 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;
 - + Removal of impediments to team's progress
 - + Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.



ScrumMaster

Software Development Methodologies

Scrum Team – Scrum Master

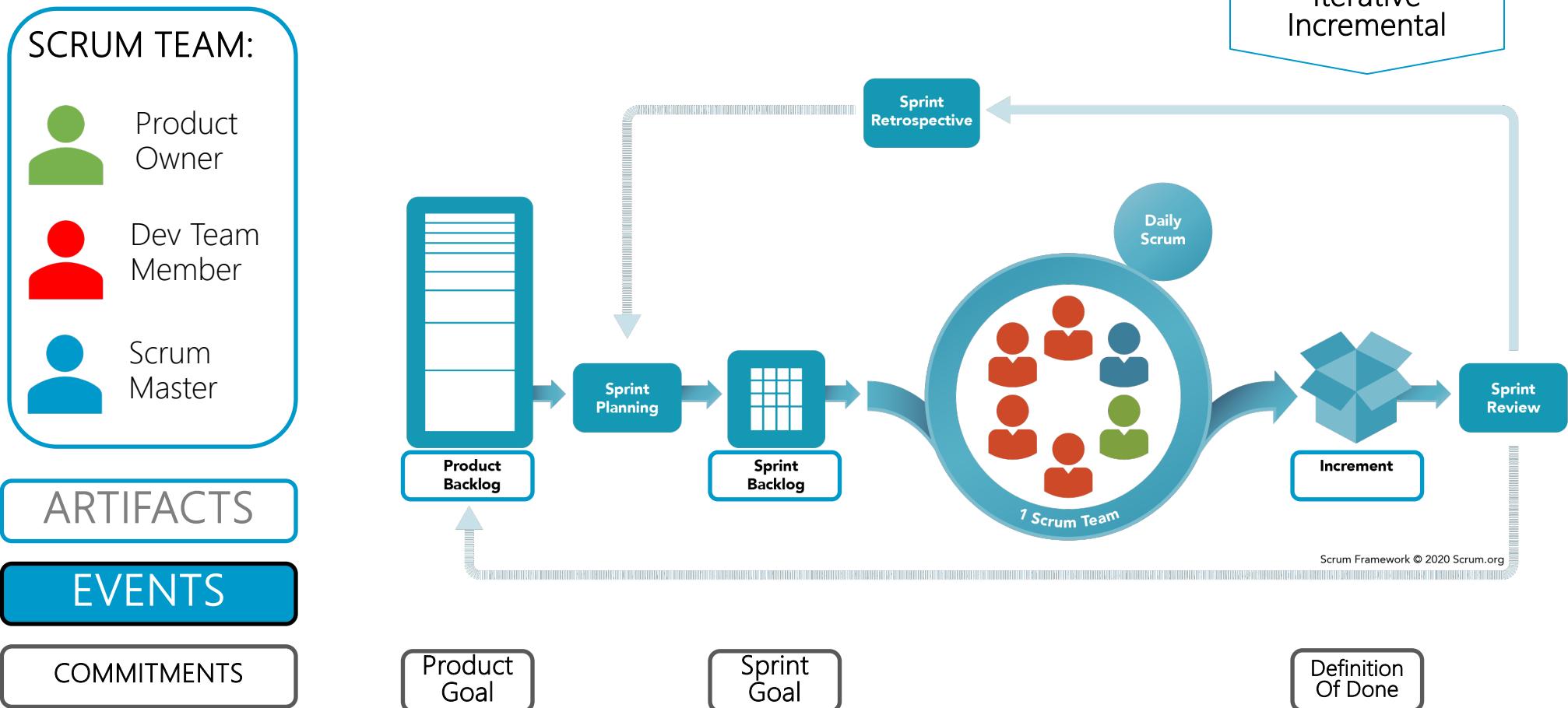
- + Is accountable for establishing Scrum as defined in the Scrum Guide, in theory and practice
- + Accountable for the Scrum Team's effectiveness
 - + Coaching the team members in self-management and cross-functionality;
 - + Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
 - + Removal of impediments to team's progress
 - + Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.



ScrumMaster

Software Development Methodologies

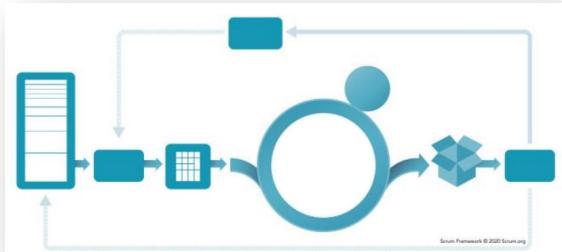
SCRUM – Process Framework - Events



Software Development Methodologies

Scrum Events – Sprint

PROCESS



ROLES



TIMEBOX

Max. 1 Month
Most Common:
2 weeks

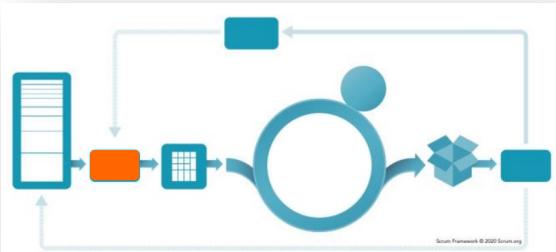
Sprints are the heartbeat of Scrum, where ideas are turned into value

- + Mother of all other events: Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective
- + 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..
- + Ensures inspection and adaptation of progress towards the Product Goal, generating learning.

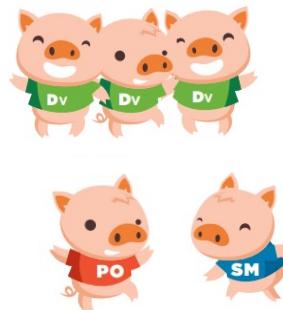
Software Development Methodologies

Scrum Events – Sprint Planning

PROCESS



ROLES



TIMEBOX

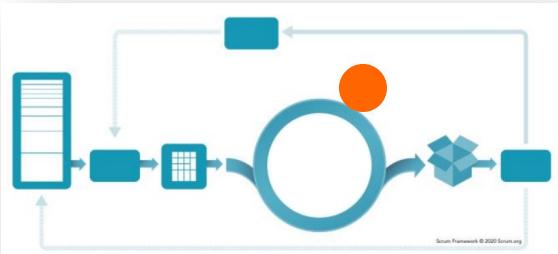
Max. 8 hours
for 4 weeks
sprint

- + **Topic #1: Why** is this sprint valuable?
 - + The **Product Owner (PO)** proposes based on a prioritized Prod. Backlog how the product could increase its value and utility.
 - + Scrum Team define Sprint Goal aligned always with the Product Goal.
- + **Topic #2: What** can be done this sprint?
 - + Discussing with the Product Owner, the **Developers** select items from the Product Backlog to include in the current Sprint.
 - + Forecast based on past performance, upcoming capacity, and Definition of Done (DoD),
- + **Topic #3: How** will the chosen work get done?
 - + For each selected Product Backlog Item (PBI), the **Developers** plan the work necessary to create an Increment that meets the Definition of Done. This is often done by decomposing PBI's into smaller work items **of one day or less**.

Software Development Methodologies

Scrum Events – Daily Scrum

PROCESS



ROLES



TIMEBOX

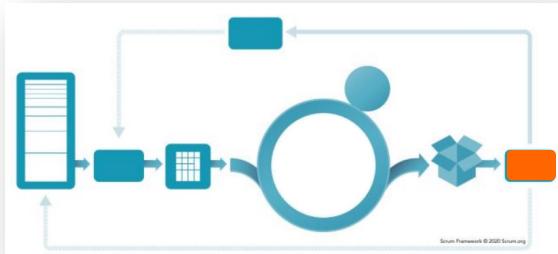
Max. 15 min
per day.
Same time and
place

- + 3 Questions:
 - + 1) What I've done yesterday?
 - + 2) What am I going to do today?
 - + 3) Is there any impediment (personal or team) to achieve Sprint Goal?
- + The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary.
- + PO and SM can participate as developers if they are actively working on items in the Sprint Backlog. No Chickens!
- + Daily Scrum produces an actionable plan for the next day of work.

Software Development Methodologies

Scrum Events – Sprint Review

PROCESS



ROLES



TIMEBOX

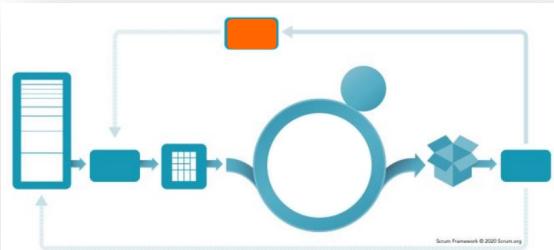
Max. **4 hours**
for 4 weeks
sprints.

- + Inspect the outcome of the Sprint and determine future adaptations.
- + The Scrum Team presents the results of their work to key stakeholders, what was accomplished in the Sprint, and progress toward the Product Goal is discussed
- + Based on this information, attendees collaborate on what to do next.
- + The Product Backlog may also be adjusted to meet new opportunities: ready to adapt for changes in product priorities.

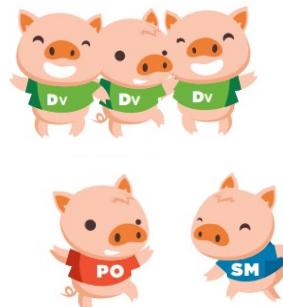
Software Development Methodologies

Scrum Events – Sprint Retrospective

PROCESS



ROLES



TIMEBOX

Max. 3 hours
for 4 weeks
sprints.

- + Plan ways to increase quality and effectiveness.
- + Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done.
- + Scrum Team discusses what went well during the Sprint, what problems it encountered, and how those problems were (or were not) solved.
- + Scrum Team identifies the most helpful changes to improve its effectiveness. The most impactful improvements are addressed as soon as possible and may be added to next Sprint Backlog.

Software Development Methodologies

Scrum Events – Example for a 2 weeks sprint

SEGUNDA-FEIRA	TERÇA-FEIRA	QUARTA-FEIRA	QUINTA-FEIRA	SEXTA-FEIRA
Sprint Planning	2	3	4	5
D.S.	D.S.	D.S.	D.S.	D.S.
8	9	10	11	12
D.S.	D.S.	D.S.	D.S.	D.S.
Product Backlog Refinement		Sprint Review		
		Sprint Retrospective		

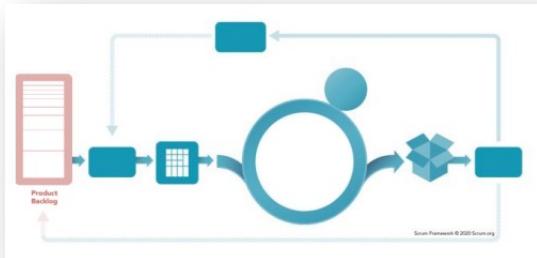


Daily Scrum Meeting

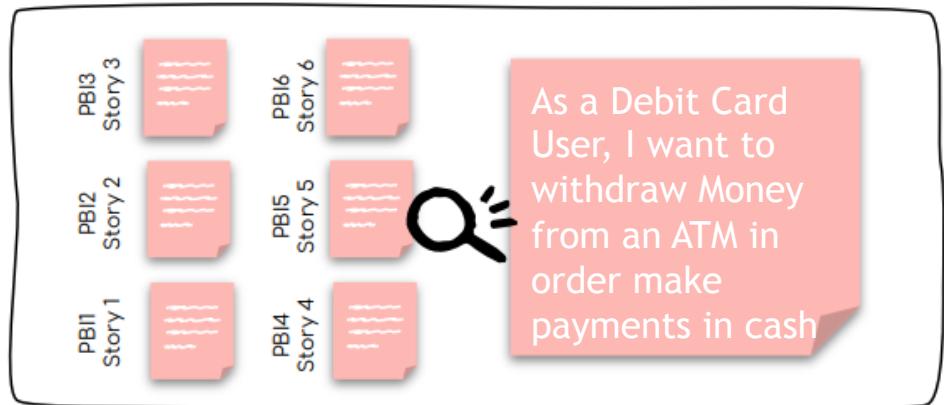
Software Development Methodologies

Scrum Artifacts – Product Backlog

PROCESS



EXAMPLE



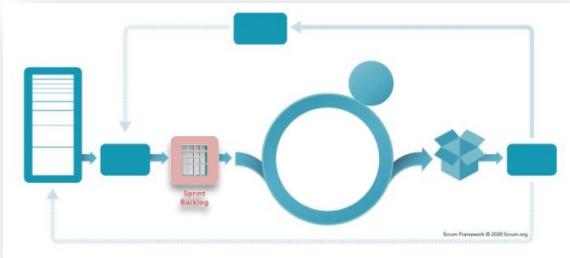
The Product Backlog is the single source of truth for the Scrum Team.”

- + The Product Backlog (PB) is an emergent, ordered list of **what** is needed to improve the product.
- + Product Backlog refinement is the act of breaking down and further defining Product Backlog Items (PBI) into smaller, more precise items. This is an ongoing activity to add details, such as a description, order, and size
- + **Commitment: 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. Is the long-term objective for the Scrum Team

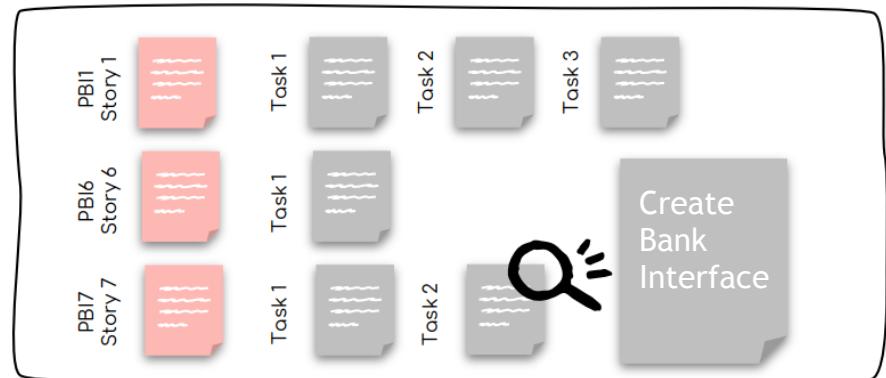
Software Development Methodologies

Scrum Artifacts – Sprint Backlog

PROCESS



EXAMPLE

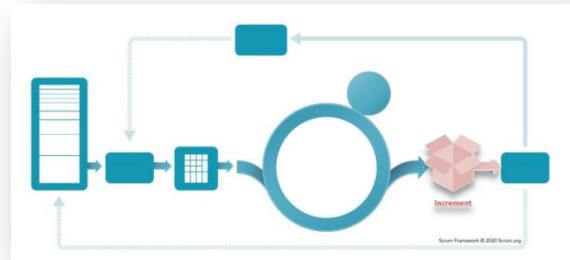


- + The Sprint Backlog is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), as well as an actionable plan for delivering the Increment (how).
- + Sprint Goal also creates coherence and focus, encouraging the Scrum Team to work together
- + **Commitment: Sprint Goal**
 - + The Sprint Goal is created during the Sprint Planning event and then added to the Sprint Backlog.
 - + 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.

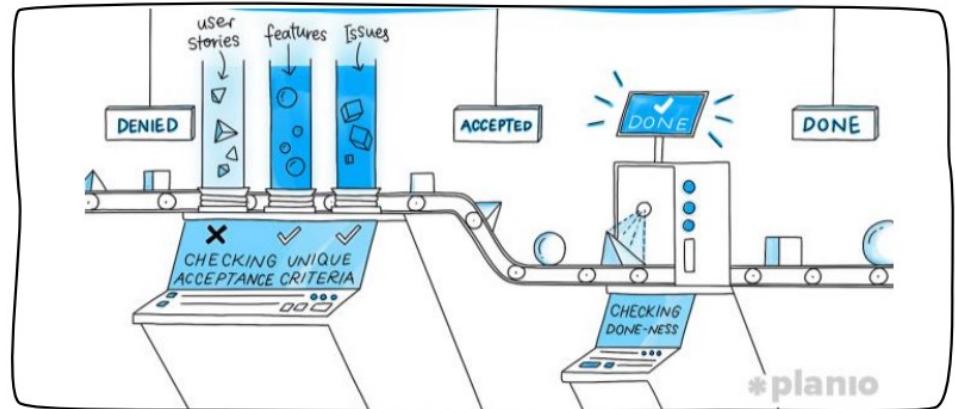
Software Development Methodologies

Scrum Artifacts – Increment

PROCESS



EXAMPLE



- + An Increment is concrete progress 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
- + **Commitment: Definition of Done**
 - + The Definition of Done is a formal description of the state of the Increment when it meets the quality gate required
 - + The moment a Product Backlog item meets the Definition of Done, an Increment is born

Software Development Methodologies

Scrum Artifacts – Definition of Done

The Product Increment MUST adhere to the current Definition of Done

Why Is This Important Even If the PO Doesn't Release the Increment?

- Manifesto Principles
 - #7. Working software is the primary measure of progress.
 - #1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
 - #3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Impact of Not Following the Current Definition of Done (DoD)
 - Trust is damaged if we deliver something of lower "quality" (purpose)
 - The vision for technical excellence (mastery) is damaged
 - Predictability is damaged (autonomy)
 - Anything not "done" equates to technical debt (quality)

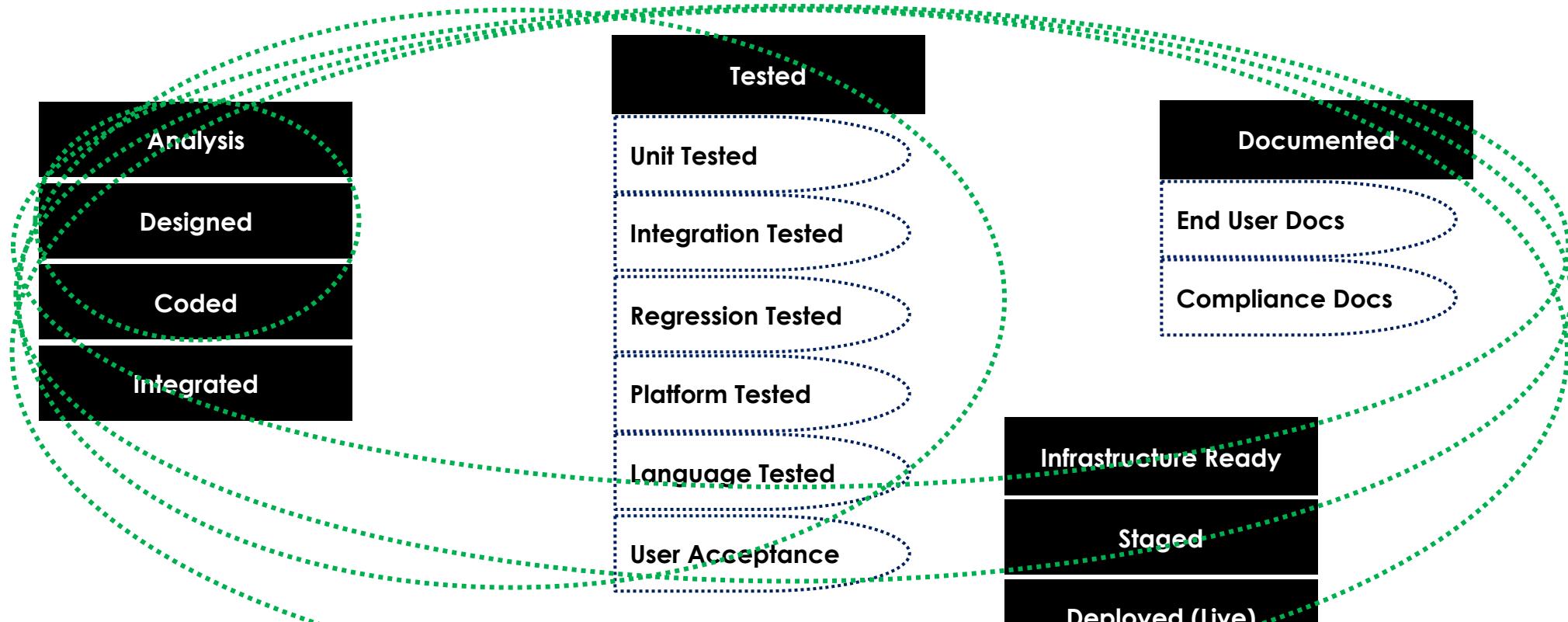
Software Development Methodologies

Scrum Artifacts – Definition of Done (DoD) - Examples

State of Confidence
<ul style="list-style-type: none">• X% of code coverage from Unit Test• Automated tests for the User Story are created and pass• Acceptance criteria is met• All regression testing of the product is passing• System-level Non-Functional requirements (e.g. System-level performance) tested and passed• Software is running in the server to be defined by the team (optimally in pre-production)• Code review by peer• Technical documentation updated• User documentation updated & localized• Localization for the story and localization testing is done• Marketing input is done• Legal documents are done• Beta Testing is done

Software Development Methodologies

Scrum Artifacts – Definition of Done (DoD) – Evolution over Time



Extend the definition as far as practical

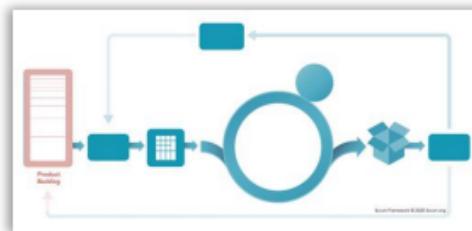
Software Architecture

Agenda

- + Software Development Methodologies:
 - + Agile
 - + Organizational framework
 - + Mindset, values & principles
 - + Scrum
 - + Process / Events / Artifacts
 - + Practices

Software Development Methodologies

Scrum Practices - Product Backlog Building



Stakeholder Management

Personas & Customer Journey

Product Roadmap

Story Maps

Release Plan

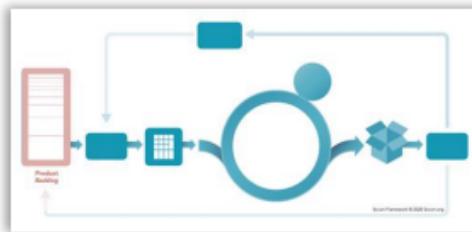
User Stories & Scenarios

FOCUS-TBD

Wireframes

Software Development Methodologies

Scrum Practices - Product Backlog Building



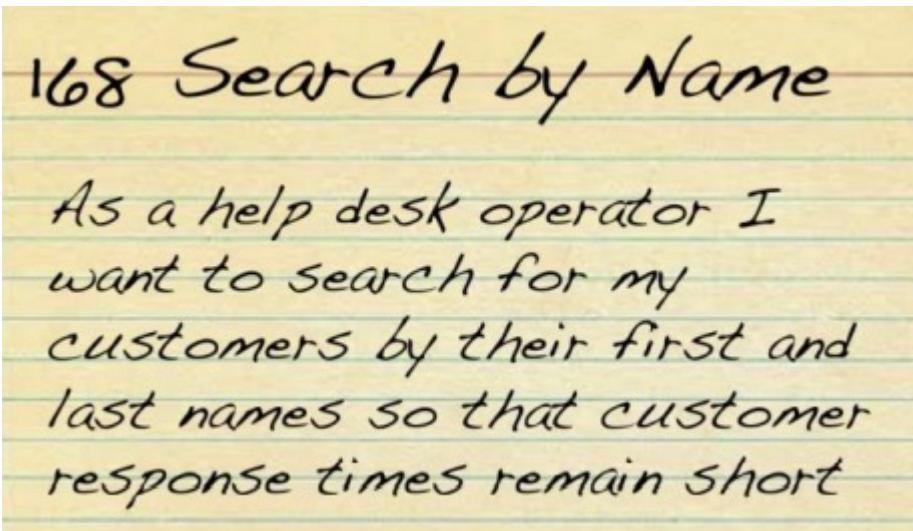
Software Development Methodologies

Scrum Practices – User Stories (US)



- + User Stories are ways to specify the features of a product, which detail a user's need and describe it from their point of view
- + They are usually written with meta-language and on cards: physical (post-its) or virtual in an agile tool (ex: jira, gitlab)

As **<role/user>**, I want to **<function/functionality>**, so that **<benefit/value>**



User Story		
Title:	Priority:	Estimate:
User Story: As a [description of user], I want [functionality] so that [benefit].		
Acceptance Criteria: Given [how things begin] When [action taken] Then [outcome of taking action]		

Software Development Methodologies

Scrum Practices – User Stories (US)



- + One of the techniques to ensure the quality of User Stories is to ensure that it has the characteristics described by the acronym **INVEST**

I ndependent	Does not depend on another User Story(ies) to be implemented
N egotiable	A story is not a contract. result needs to be the result of collaborative negotiation between the customer (or Product Owner) and the developer team
V aluable	If a story does not have discernable value to the product / customer it should not be done. User Stories must be prioritized in the backlog according to business value
E stimable	A story must be able to be estimated or sized so it can be properly prioritized. A US with high value but extremely lengthy dev time may not be the highest priority item
S mall	For two week sprints user stories should have in average 3-4 days of work in TOTAL! This includes all work to get the story to a “done” state
T estable	Every story needs to be testable in order to be “done”.Testable meaning acceptance criteria can be written immediately

Software Development Methodologies

Scrum Practices – User Stories (US) & Scenarios

User Stories & Scenarios

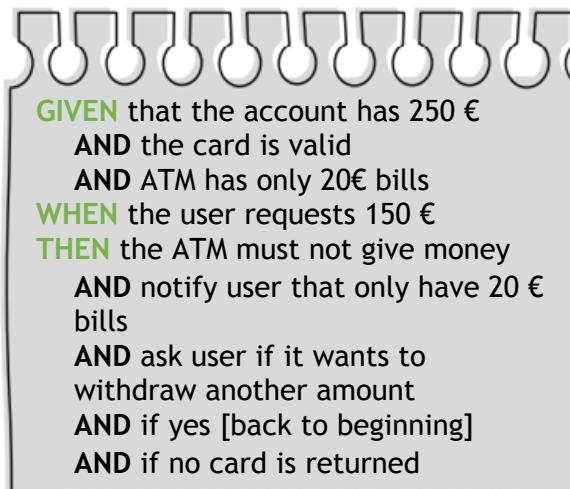
- + Scenarios allow you to create possibilities and define acceptance criteria:
 - + 1st scenario: the ideal scenario, where the goal of User Story is achieved without challenges
 - + 2nd scenario: alternative scenarios where the objective is also achieved
 - + 3rd scenario: scenarios where the goal is not achieved

Given **<Context>**, When **<action>**, then **<result>**

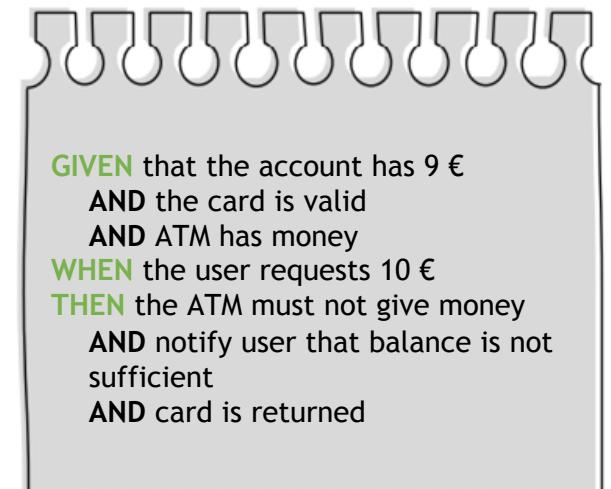
1st scenario: Account has enough balance



2nd scenario: ATM Only has 20€ bills



3rd scenario: account does not have enough balance



Software Development Methodologies

Scrum Practices – User Stories (US) & Scenarios – Who writes USs?



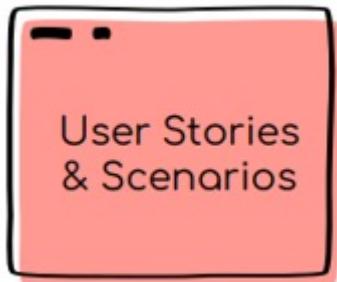
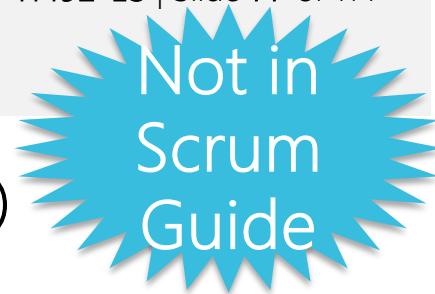
The Scrum Guide does not state who must write user stories or other PBIs

Product Owner	Development Team
<ul style="list-style-type: none">• Provide clarification• Collaborate with the Dev Team	<ul style="list-style-type: none">• Ask questions about the PBIs• Collaborate with the PO• Estimate PBIs
ScrumMaster	Subject Matter Experts
<ul style="list-style-type: none">• Schedule & facilitate backlog refinement sessions (optional)• Follow up on impediments raised during sessions• Help the PO and DTM discover better ways to refine the product backlog	<ul style="list-style-type: none">• Provide clarification• Offer advice to the Dev Team Members (DTM) based on their experience

3 Cs
CARD | CONVERSATION | CONFIRMATION

Software Development Methodologies

Scrum Practices – User Stories (US) – Definition of Ready (DoR)



- Use a “Definition of Ready” that has been agreed to by the team to validate that the user story is “ready”
- The Definition of Ready might include:
 - INVEST
 - Acceptance Criteria (in a specific format like Given-When-Then)
 - Estimated in Story Points
 - Prioritization ranked appropriately in the product backlog
- Think of stories from the perspective of delivering value, not tasks
- Refinement usually consumes no more than 10% of the capacity of the Development Team.

Software Development Methodologies

Scrum Practices – User Stories (US) – DoR vs DoD



Definition of Ready

- Describes what, not how
- Independent
- Negotiable
- Valuable
- Estimable
- Stories are small enough
- Testable
- Delivers business value
- Requirement understood by Development Team
- Acceptance Criteria agreed

Prep

Make the
DoR & DoD
visible – put
them on the
team board

Definition of Done

- X% of code coverage from Unit Test
- Automated tests for the User Story are created and pass
- Acceptance criteria is met
- All regression testing of the product is passing
- System-level Non-Functional requirements (e.g. System-level performance) tested and passed
- Software is running in the server to be defined by the team (optimally in pre-production)
- Code review by peer
- Technical documentation updated
- User documentation updated & localized
- Localization for the story and localization testing is done
- Marketing input is done
- Legal documents are done
- Beta Testing is done

Validation

Software Development Methodologies

Scrum Practices – User Stories (US)



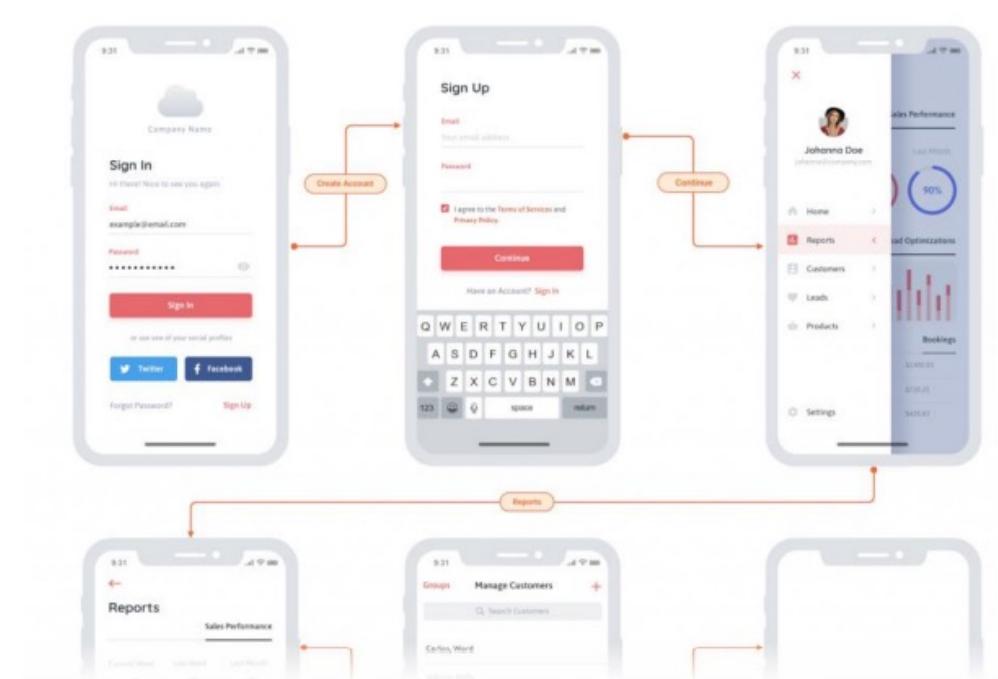
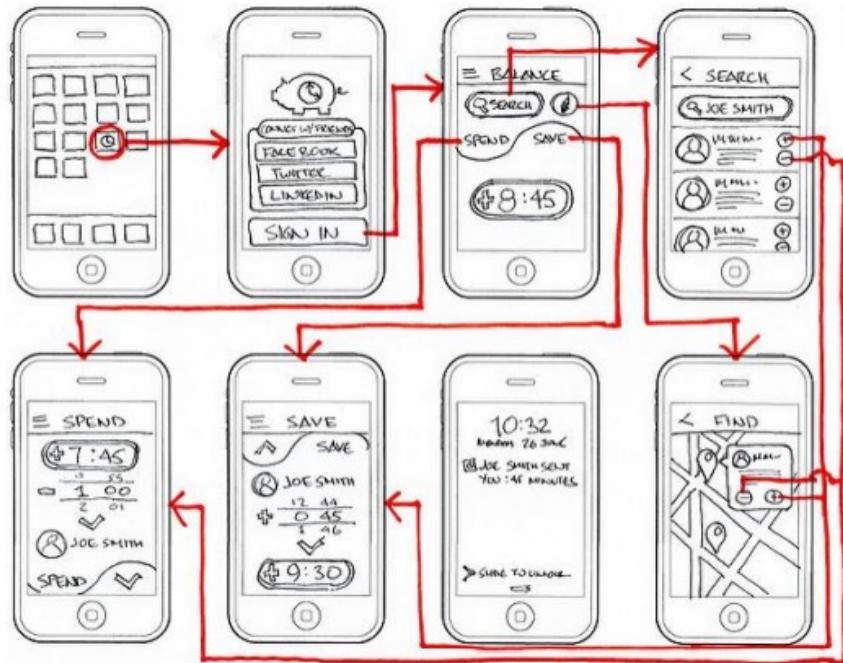
- Use a “Definition of Ready” that has been agreed to by the team to validate that the user story is “ready”
- The Definition of Ready might include:
 - INVEST
 - Acceptance Criteria (in a specific format like Given-When-Then)
 - Estimated in Story Points
 - Prioritization ranked appropriately in the product backlog
- Think of stories from the perspective of delivering value, not tasks
- Refinement usually consumes no more than 10% of the capacity of the Development Team.

Software Development Methodologies

Scrum Practices – Wireframes

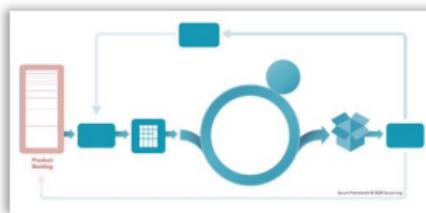


- + Wireframes of an application's screens or navigation streams. They can be done manually or in specific tools.
- + They allow in a quick and inexpensive way to present to stakeholders how the final product will be.



Software Development Methodologies

Scrum Practices – Product Backlog Priorization



Modelo
MoSCoW

Matriz de
Eisenhower

Kano
Analysis

Modelo
Wiegers

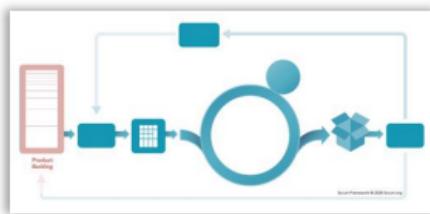
Multivoting

Backlog
Priorizado

Risk Adjusted
Backlog

Software Development Methodologies

Scrum Practices – Product Backlog Priorization



Modelo
MoSCoW

Multivoting

Backlog
Priorizado

Software Development Methodologies

Exercise – Product Backlog Priorization

Celebrity Prioritisation

THE SCENE. There is a cruise-liner with a number of famous people on-board who are now in mortal danger as the boat has hit an iceberg and is sinking. We don't know how long it will be until the boat has sunk but the good news is that we have a rescue boat and can be heroes by rescuing them. The bad news is that our rescue boat is small and we can only rescue one person at a time.

GROUP EXERCISE. As a team, put the celebrities (on next slide) in the order in which you would rescue them, Given that they have the most value to our society / human race.

TIME-BOX. 10 minutes

Software Development Methodologies

Exercise – Product Backlog Priorization (2)



a)



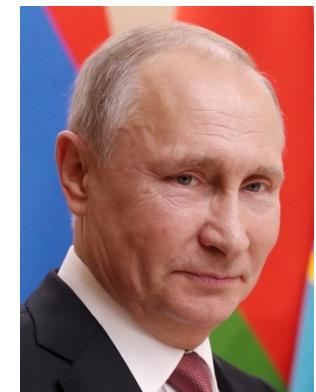
b)



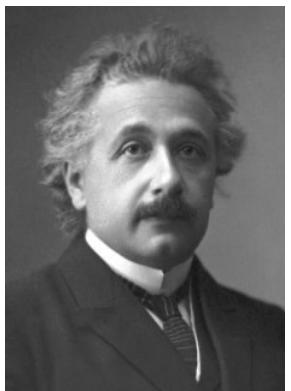
c)



d)



e)



f)



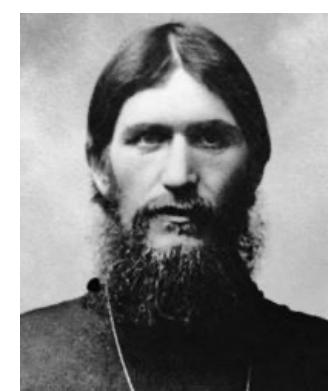
g)



h)



i)



j)

Software Development Methodologies

Exercise – Product Backlog Priorization

Celebrity Prioritisation

Task #2: Using the scale (0, 1, 2, 3, 5, 7, 13, 21, 40) estimate how useful each celebrity is to the rescue effort.

0 = they are a walking survival package

40 = there is no hope

Task #3: Arrange the celebrities into a big boat. However, the boat only has 20 of rescue effort capacity

TIME-BOX. 10 minutes

Software Development Methodologies

Scrum Practices – Product Backlog Priorization

Modelo
MoSCoW

- + Allows to determine the level of impact of the non-realization of each functionality and thus prioritize by importance for the desired end result

Or Otherwise, the solution:

MUST

HAVE -----> does not go live!

O

SHOULD

HAVE IF VIABLE -----> will have serious functional restrictions

COULD

HAVE IF DOES NOT HAVE
A NEGATIVE IMPACT -----> will be incomplete but fully functional. Only
advanced users will notice.

O

WON'T

HAVE NOW BUT WILL HAVE
IN THE FUTURE -----> will be fully functional and no one will notice
(nice to have)

Software Development Methodologies

Scrum Practices – Product Backlog Priorization

Multivoting

- + Decision method that transforms a large number of options into a priority list or a final selection.
- + Allows for options, that aren't personal preference but is at the top of the preferences of many, can be still selected as priorities

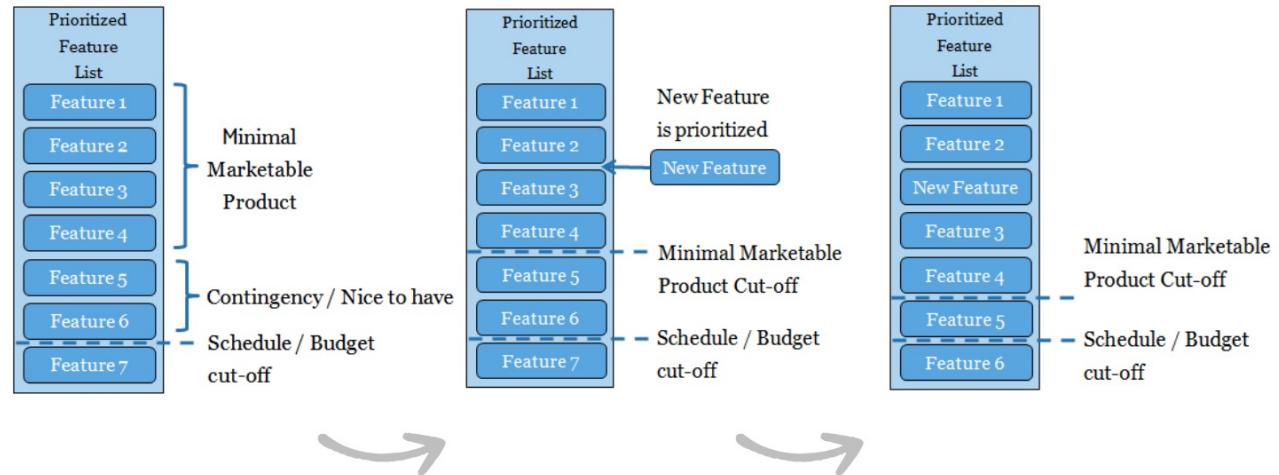


Software Development Methodologies

Scrum Practices – Product Backlog Priorization

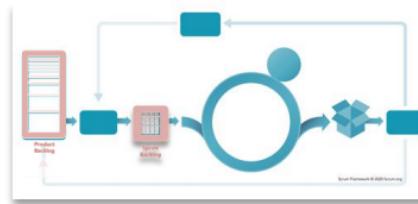
Backlog Priorizado

- + Phased prioritization technique, where priority decisions are made considering restrictions and changes.
- + **STEP #1:** Make the feature list realistic by including contingencies. The **budget** and **time objective** limits the scope.
- + **STEP #2:** Manage issues and changes. Whenever they appear, you may have to drop a less priority feature



Software Development Methodologies

Scrum Practices – Product & Sprint Backlog Estimation



Affinity
Estimating
(High-level)

Story Points

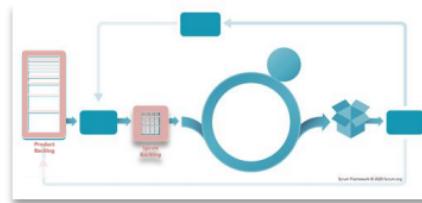
Wideband
Delphi

Planning
Poker

Affinity
Estimating
(User story)

Software Development Methodologies

Scrum Practices – Product & Sprint Backlog Estimation



Affinity
Estimating
(High-level)

Story Points

Planning
Poker

Software Development Methodologies

Scrum Practices – Product & Sprint Backlog Estimation

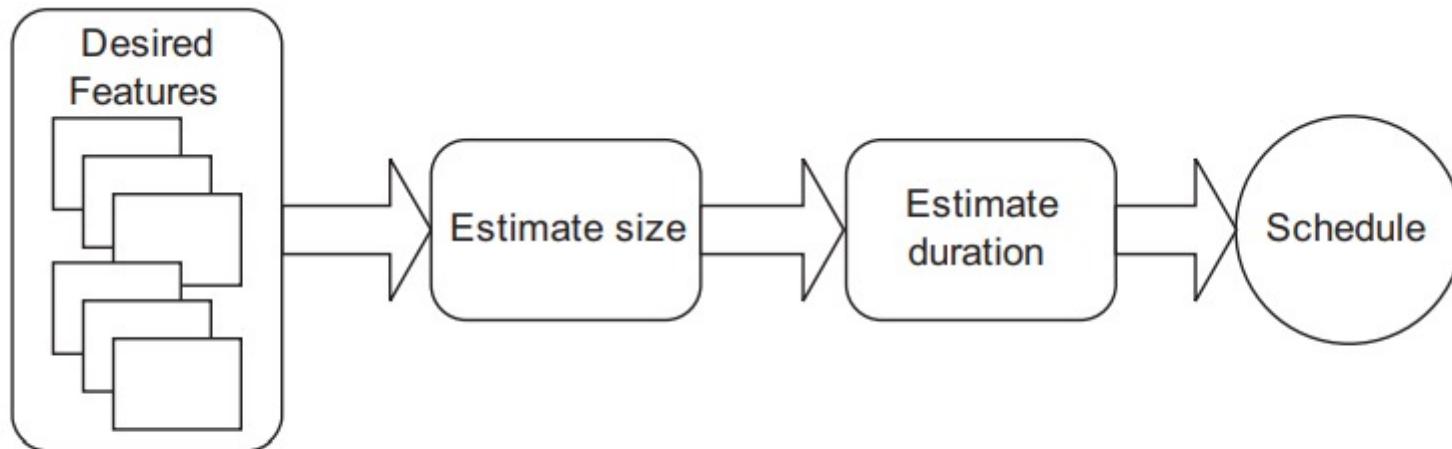
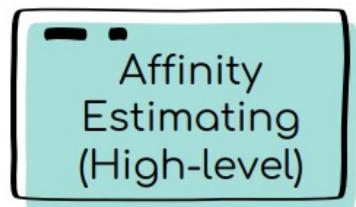


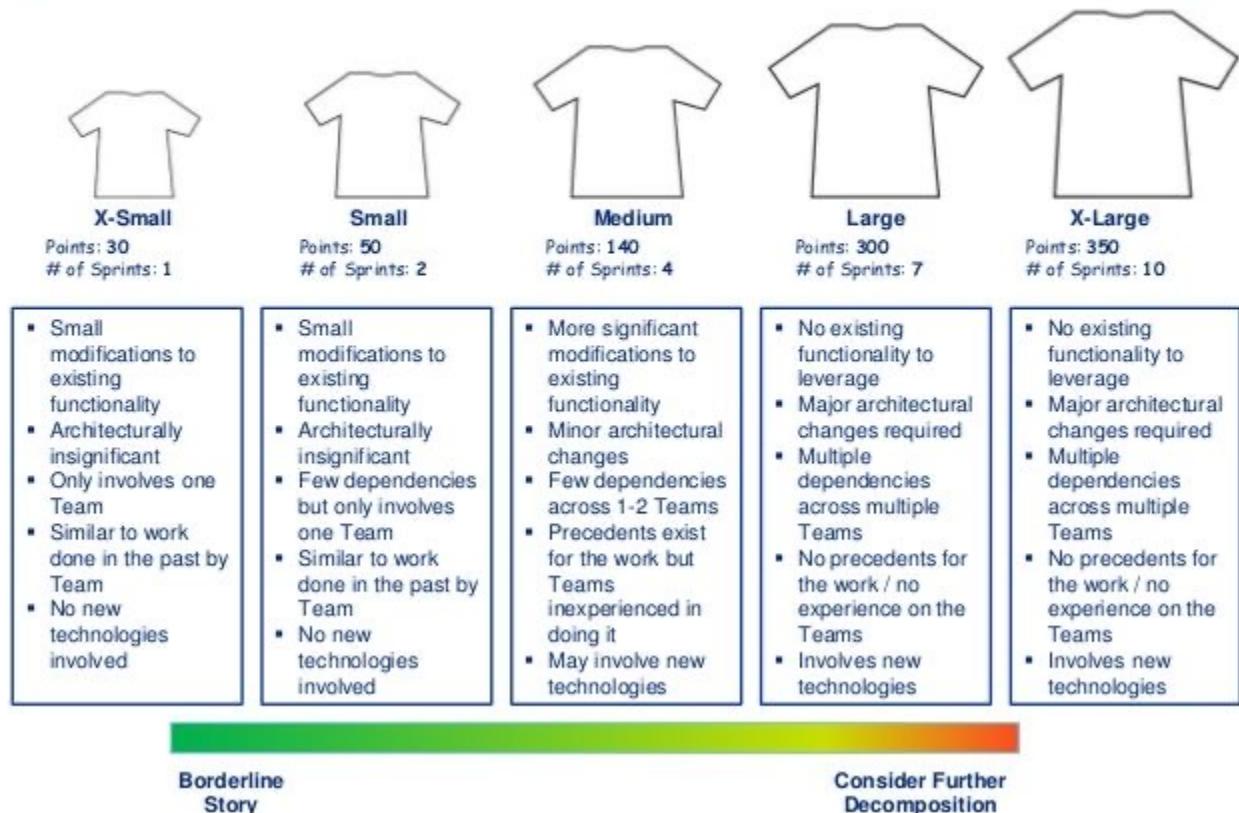
Figure II.1 Estimating the duration of a project begins with estimating its size.

Software Development Methodologies

Scrum Practices – Product & Sprint Backlog Estimation



- + Relative estimate that uses the size of t-shirts or buckets as a reference of the size of the work to be performed.
- + May apply to estimation for Sprint, Releases, or project sizing in portfolio management.



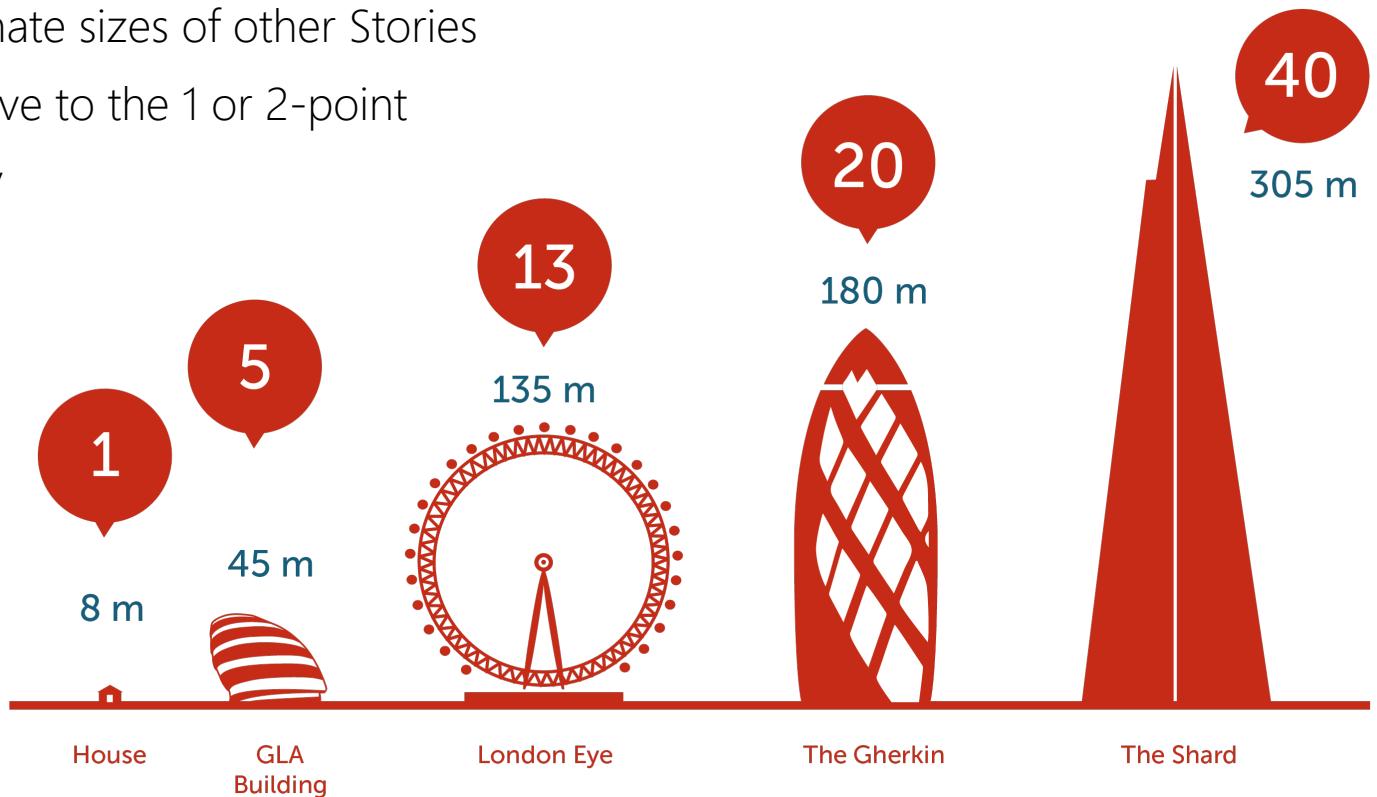
Software Development Methodologies

Scrum Practices – Product & Sprint Backlog Estimation

Story Points

- + Identify simplest Story and give it a size of 1 or 2 points
- + Estimate sizes of other Stories relative to the 1 or 2-point Story

- + What about 0 (zero) points User Stories?



Software Development Methodologies

Scrum Practices – Product & Sprint Backlog Estimation



- + Group technique based on consensus. Generates better estimates by gathering different perspectives and stimulating the why of each evaluation.
- + Estimates are then fine-tuned, and aspects not considered until then appear. The unit of measure is the Story Points

- Each player (developer) has a set of cards
 - E.g. 1,2,3,5,8,13,20,40,100, 200, infinity (Modified Fibonacci sequence)
- Product Owner explains story
- Scrum Master facilitates so that all of the developers understand & participate
- Each player, at the same time, shows their estimate card
- If there are “outliers” in the estimates, that person explains
- Team votes again
- If there are still outliers, the facilitator identifies the average score closest to the nearest modified Fibonacci sequence
- Facilitator records the story point value
- Repeat

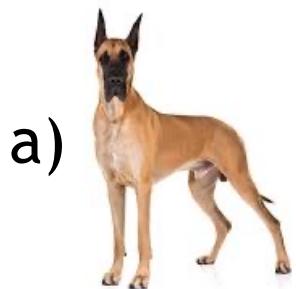


The
conversation
is more
important
than the
estimate!

Software Development Methodologies

Exercise – Dog Show – Smallest Effort

Prepare these dogs for a dog show!



a)



b)



c)



d)



e)



f)



g)



h)

- pick the 1 dog that represents the smallest effort and then give that dog a “1” .
- Relative to that dog, estimate how much effort is required to groom each dog in preparation for the dog show.
- Training is not part of “done” .

Software Development Methodologies

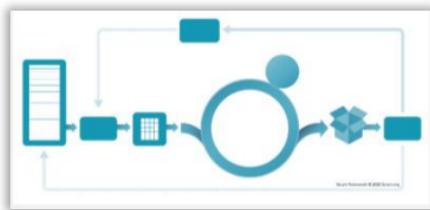
Exercise – Dog Show – Smallest Effort

- **Great Dane** – very sweet dog. Short hair, but huge.
- **Chihuahua** – the meanest dog you are ever going to try to bathe.
- **German Shepherd** – sweet dog but really hairy.
- **Dachshund** – just wants to play. Short hair.
- **Collie** – immense amount of hair.
- **2 Dalmatian Puppies** – scared because of all of the other dogs around them. 1st show. 2 of them together. Short hair but have been rolling around in mud all day.
- **Unknown breed** – ferocious, long hair, huge (might eat you)
- **Doberman** – docile, short hair



Software Development Methodologies

Scrum Practices – Work Flow Management



Gestão do fluxo de trabalho

Burn Down & Up Charts

Velocidade

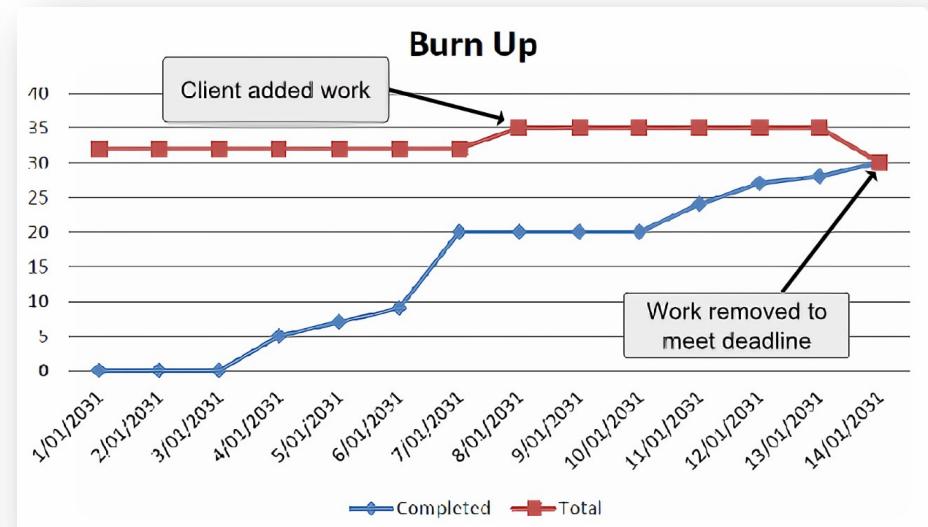
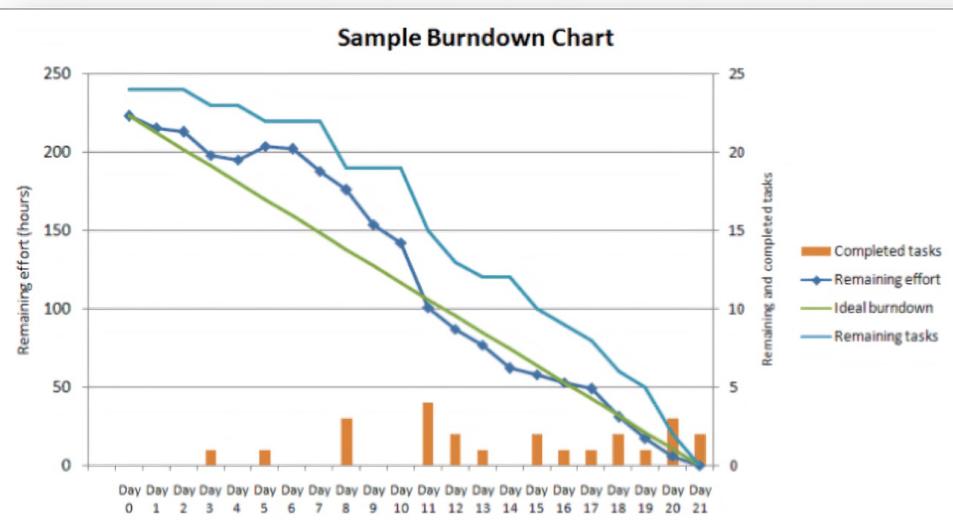
Kanban Boards

Software Development Methodologies

Scrum Practices – Work Flow Management

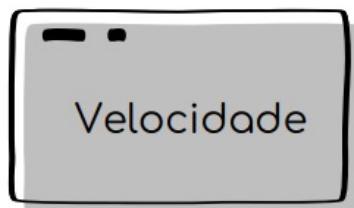
Burn Down & Up Charts

- + Visual representations of work (story points) versus expectation.
- + The Burn-Down chart compares the amount of work to be done with a linear trend.
- + The Burn-Up Chart compares the amount of work done with the sprint scope.

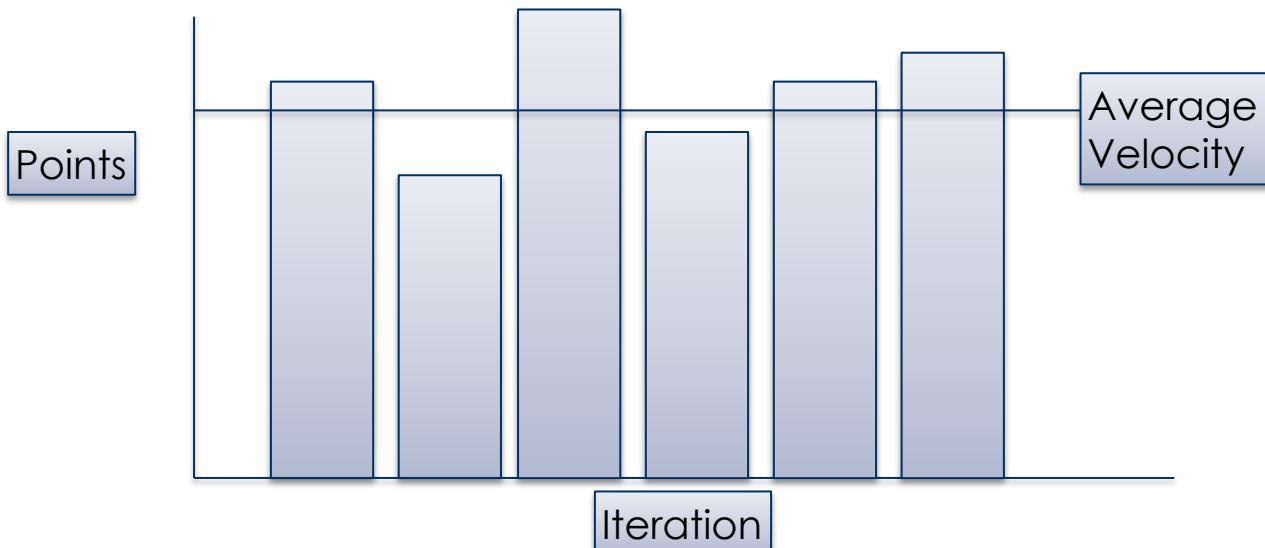


Software Development Methodologies

Scrum Practices – Work Flow Management



- + Amount of Story Points that a team can perform in a certain period of time, serving as a reference to estimate future work.
- + Knowing Speed and using estimation techniques like Yesterday Weather will be easier to predict the execution capability in an upcoming Sprint.

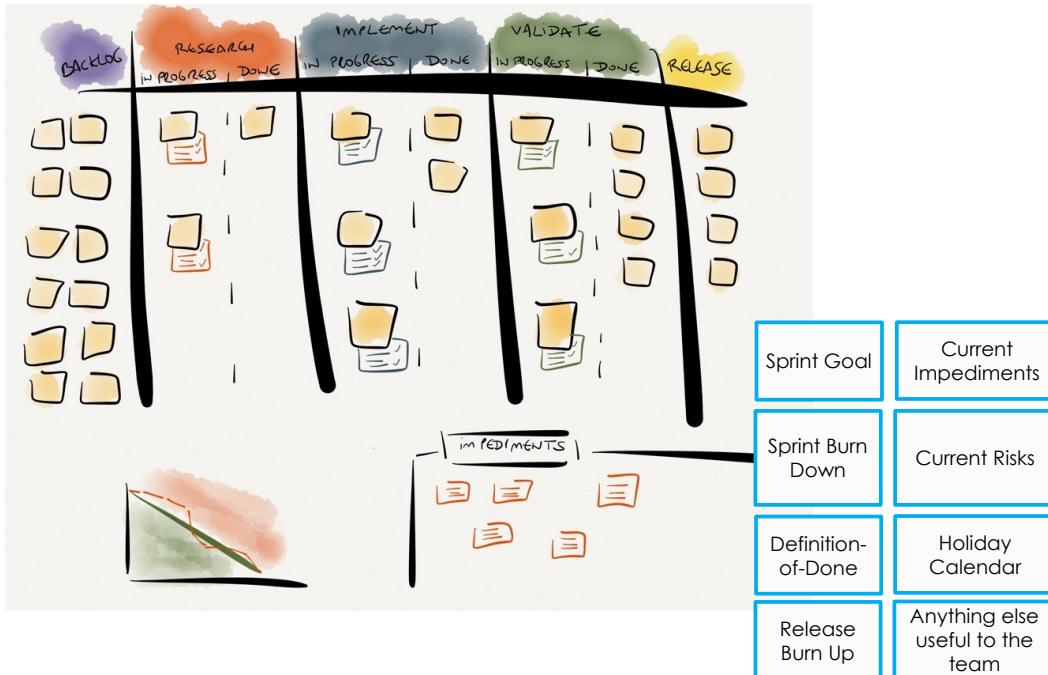


Software Development Methodologies

Scrum Practices – Work Flow Management



- + Based on industrial processes, it was adopted by Agile for its ability to communicate and minimize waste.



"LOW TECH, HIGH TOUCH"

Intentionally simple, it has the advantages of:

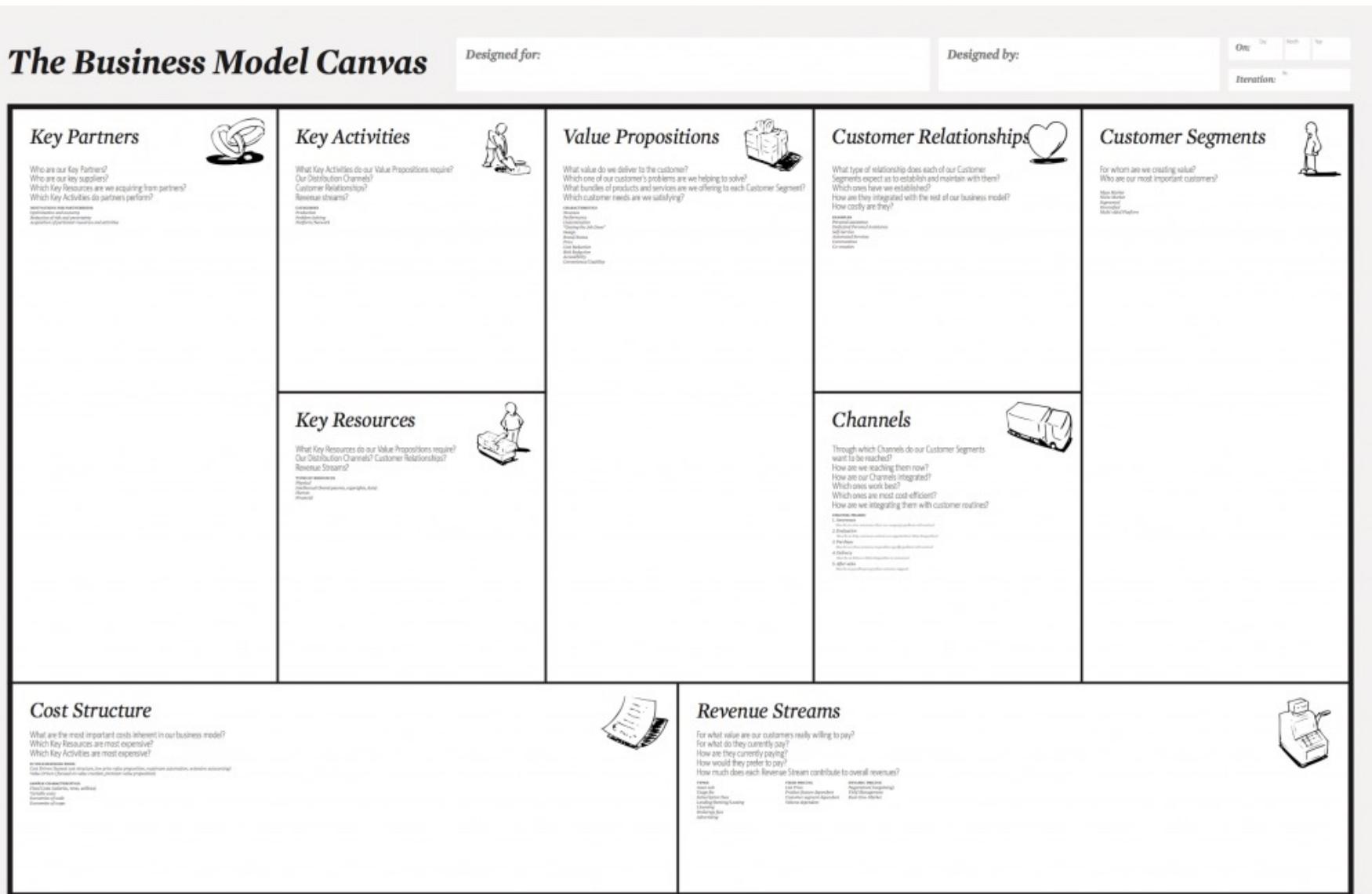
- Facilitate common understanding of what it is to do
- Promote interaction between all stakeholders in the discussion and in plan changes
- Allow the view of the entire workflow, and tasks advance through a “pull system”
- Facilitar a análise e melhoria dos processos
- Minimizar o Work in Progress (WIP)

Software Development Methodologies

Recommended Reading

Agile Manifesto & 12 Principles	Scrum Guide
<p>The Manifesto for Agile Software development consists of 1 page. The Agile Manifesto' s accompanying 12 principles consist of 2 pages.</p> <p>Agile Manifesto http://agilemanifesto.org</p> <p>12 Principles http://agilemanifesto.org/principles.html</p>	<p>The Scrum Guide is a 19-page document written by the creators of the Scrum Framework, Jeff Sutherland & Ken Schwaber.</p> <p>Scrum Guide https://www.scrumalliance.org/learn-about-scrum/the-scrum-guide</p>
<p>Agile Product Ownership in a Nutshell Video By Henrik Kniberg https://www.youtube.com/watch?v=502ILHjX9EE</p>	<p>Scrum Alliance' s Scrum eLearning Videos https://www.scrumalliance.org/learn-about-scrum/scrum-elearning-series</p>

Software Development Methodologies



Software Development Methodologies

Business Model Canvas

1. Key Partners: The strategic relationships your business creates with other companies or people.

2. Key Activities: Activities or tasks that are integral to operating your company.

3. Key Resources: Assets that are required to operate and deliver your company's value proposition.

4. Value Proposition: The fundamental need that your company is trying to fulfill for its customers. Why your company exists.

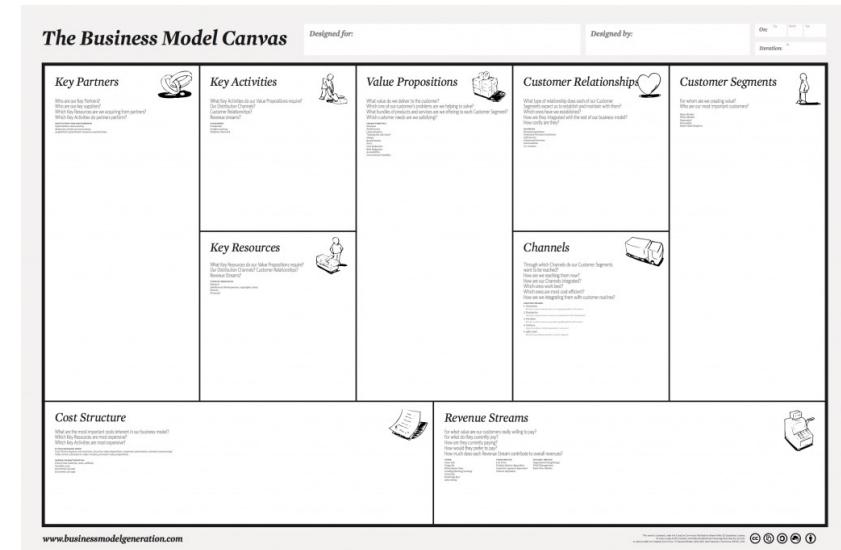
5. Customer Relationships: The type of interactions your company has with its customers and the level of support it gives.

6. Channels: Different methods that your company uses to deliver its products and value proposition to customers.

7. Customer Segments: The different groups of customers that your company interacts with.

8. Cost Structure: How a company spends money on operations. The key costs and level of cost-focus for your company.

9. Revenue Streams: Your company's sources of cash flows.



<https://vimeo.com/78350794>

Software Development Methodologies

BUSINESS MODEL CANVAS - NETFLIX

<p>● KEY PARTNERS</p> <p>Internet service providers (ISP)</p> <p>3rd party studios</p>	<p>● KEY ACTIVITIES</p> <p>Content procurement</p> <p>Application development</p> <p>3rd party licensing</p>	<p>● VALUE PROPOSITIONS</p> <p>On-demand video</p> <p>Huge selection of content</p> <p>Original content</p> <p>Competitive price point</p>	<p>● CUSTOMER RELATIONSHIPS</p> <p>Self-service platform</p>	<p>● CUSTOMER SEGMENTS</p> <p>Mass market</p>
<p>● KEY RESOURCES</p> <p>Streaming rights</p> <p>Internet bandwidth</p> <p>Recommendation algorithm</p> <p>Content library</p>			<p>● CHANNELS</p> <p>Website</p> <p>App store</p> <p>Affiliate partners</p>	
<p>● COST</p> <p>In-house content production</p> <p>3rd party licensing</p> <p>Streaming application (staff, maintenance, etc.,)</p>		<p>● REVENUE STREAMS</p> <p>Subscriptions</p>		

Software Development Methodologies

BUSINESS MODEL CANVAS - ZARA

KEY PARTNERS Providers Holding company (Inditex)	KEY ACTIVITIES Design Manufacturing Retail process (point of sale & 3rd party management) Distribution channels and logistics	VALUE PROPOSITIONS Fashionable clothes Accessories Great eCommerce experience	CUSTOMER RELATIONSHIPS Salesperson at store Brand through social media Sentimental attachment to clothing/ accessories	CUSTOMER SEGMENTS Men Women Children
KEY RESOURCES Stock Large network of stores Strong brand Logistics and supply chain infrastructure	CHANNELS Flagship store experience Fast-fashion	CHANNELS Direct store Online Social media		
COST Fixed (rent, payroll, etc.) Variables associated with sale of goods		REVENUE STREAMS Sales of clothing and accessories		

Software Development Methodologies

Business Model Canvas – Udemy V2.0

The Business Model Canvas



Software Development Methodologies

Udemy V2.0 – User Stories

Title: User Registration	Priority: High	Estimate: 3 Points
------------------------------------	--------------------------	------------------------------

As a new user, I want to be able to register for an account, so that I can access the e-Learning platform and start learning.

Acceptance criteria:

Given a new user, **when** they visit the registration page and provide valid information, **then** they should be able to create an account successfully and receive a confirmation email.

Title: User Login	Priority: Highest	Estimate: 2 Points
-----------------------------	-----------------------------	------------------------------

As a registered user, I want to log in with my credentials, so that I can access my personalized learning content and track my progress.

Acceptance criteria:

Given a **registered** user, **when** they enter their valid credentials (email and password), **then** they should be able to log in successfully and access their account.

Software Development Methodologies

Udemy V2.0 – User Stories

Title:

Course Search

Priority:

Medium

Estimate:

5 Points

As a logged-in user, I want to search for courses using keywords, so that I can quickly find relevant courses and explore my interests.

Acceptance criteria:

Given a logged-in user, **when** they search for a course by entering keywords, **then** they should see a list of relevant courses, including course titles and descriptions.

Title:

Enroll in a Course

Priority:

Highest

Estimate:

4 Points

As a logged-in user, I want to enroll in courses easily, so that I can start learning new topics and skills without any hassle.

Acceptance criteria:

Given a logged-in user, **when** they find a course they want to enroll in, **then** they should be able to click an "Enroll" button and become registered for the course.

Software Development Methodologies

Udemy V2.0 – User Stories

Title: Quizzes	Priority: Low	Estimate: 6 Points
--------------------------	-------------------------	------------------------------

As a user enrolled in a course, I want to take quizzes and receive immediate feedback, so that I can assess my understanding and improve my knowledge.

Acceptance criteria:

Given a user enrolled in a course, **when** they access a quiz, **then** they should be able to answer questions one by one, see their progress, and receive immediate feedback on their answers.

Title: View Course Progress	Priority: Medium	Estimate: 3 Points
---------------------------------------	----------------------------	------------------------------

As a user enrolled in a course, I want to view my course progress, so that I can keep track of completed modules and quizzes and plan my learning accordingly.

Acceptance criteria:

Given a user enrolled in a course, **when** they access the course page, **then** they should be able to see their progress, including completed modules and quizzes.

Software Development Methodologies

Udemy V2.0 – User Stories

Title: User Profile	Priority: Low	Estimate: 4 Points
-------------------------------	-------------------------	------------------------------

As a logged-in user, I want to access and edit my profile information, so that I can keep my personal details up-to-date and customize my learning experience.

Acceptance criteria:

Given a logged-in user, **when** they access their profile, **then** they should be able to view and edit their personal information, such as name, email, and profile picture.

Title: Admin Dashboard	Priority: High	Estimate: 8 Points
----------------------------------	--------------------------	------------------------------

As an admin user, I want access to an admin dashboard, so that I can efficiently manage courses, users, and gain insights into platform usage.

Acceptance criteria:

Given an **admin user**, **when** they log in, **then** they should be able to access an admin dashboard with CRUD features for managing courses and users, and viewing analytics about users activity (number, average session time) and courses usage.

Udemy V2.0 – User Stories

Title:

Course Creation

Priority:

Highest

Estimate:

7 Points

As an admin user, I want to create new courses so that I can provide diverse learning opportunities for users.

Acceptance criteria:

Given an **admin** user, **when** they create a new course, **then** they should be able to add modules, **quizzes**, and course content with titles, descriptions, and multimedia elements.

Title:

User Notifications

Priority:

Medium

Estimate:

4 Points

As a user, I want to receive notifications about updates and important information, so that I can stay informed and engaged with the platform.

Acceptance criteria:

Given a user, **when** there are updates or notifications (e.g., new content available, quiz results), **then** the user should receive notifications and be able to view them in their account.