Scrum

What is Scrum?

Scrum is a framework for developing and sustaining complex products.

Who invented Scrum?

Scrum framework is *developed and sustained by Ken Schwaber and Jeff Sutherland*.

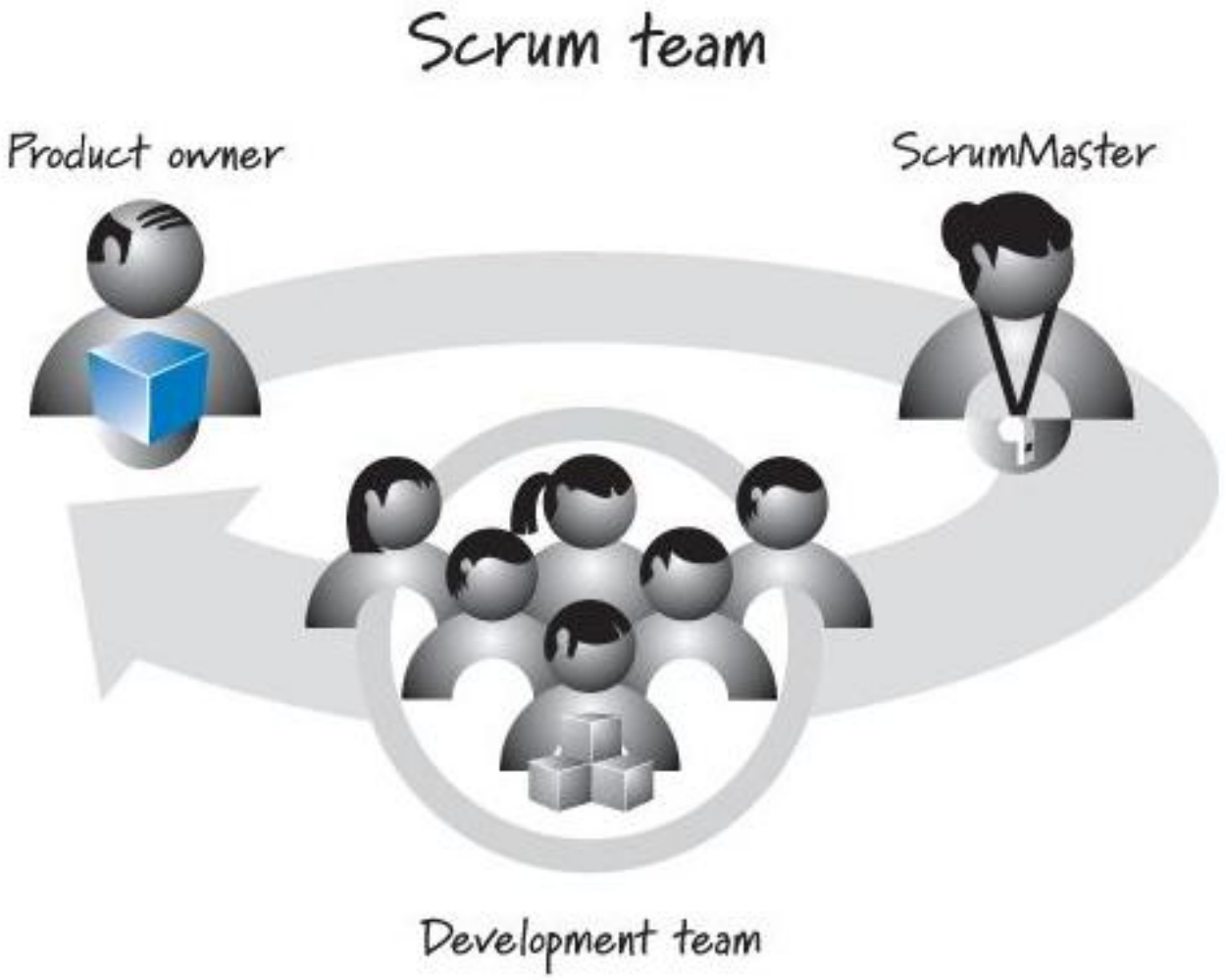
Scrum Theory

Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience *and making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk.*

Introduction

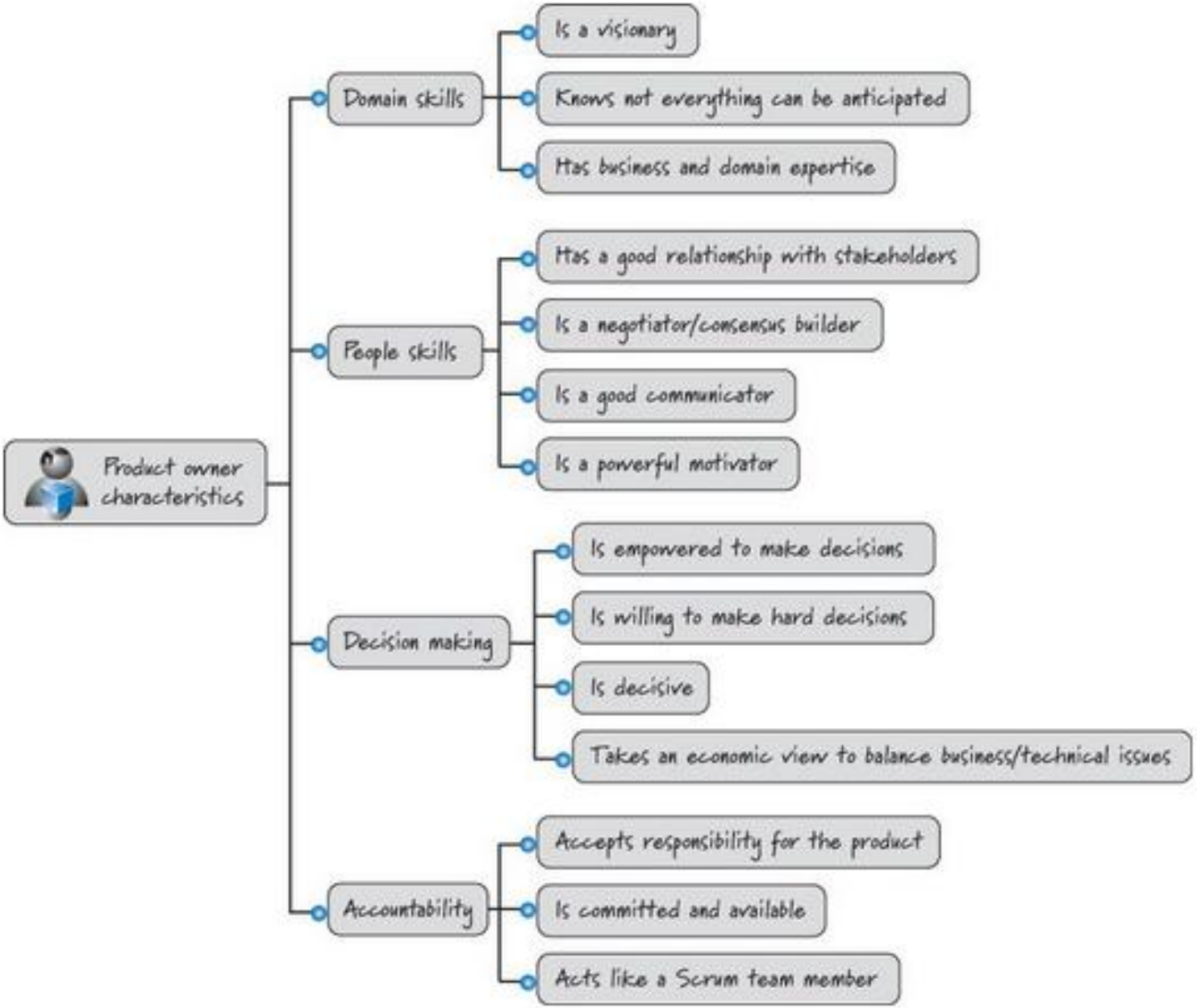
- Scrum base pillars
  - Transparency
  - Inspection
  - Adoption

Introduction

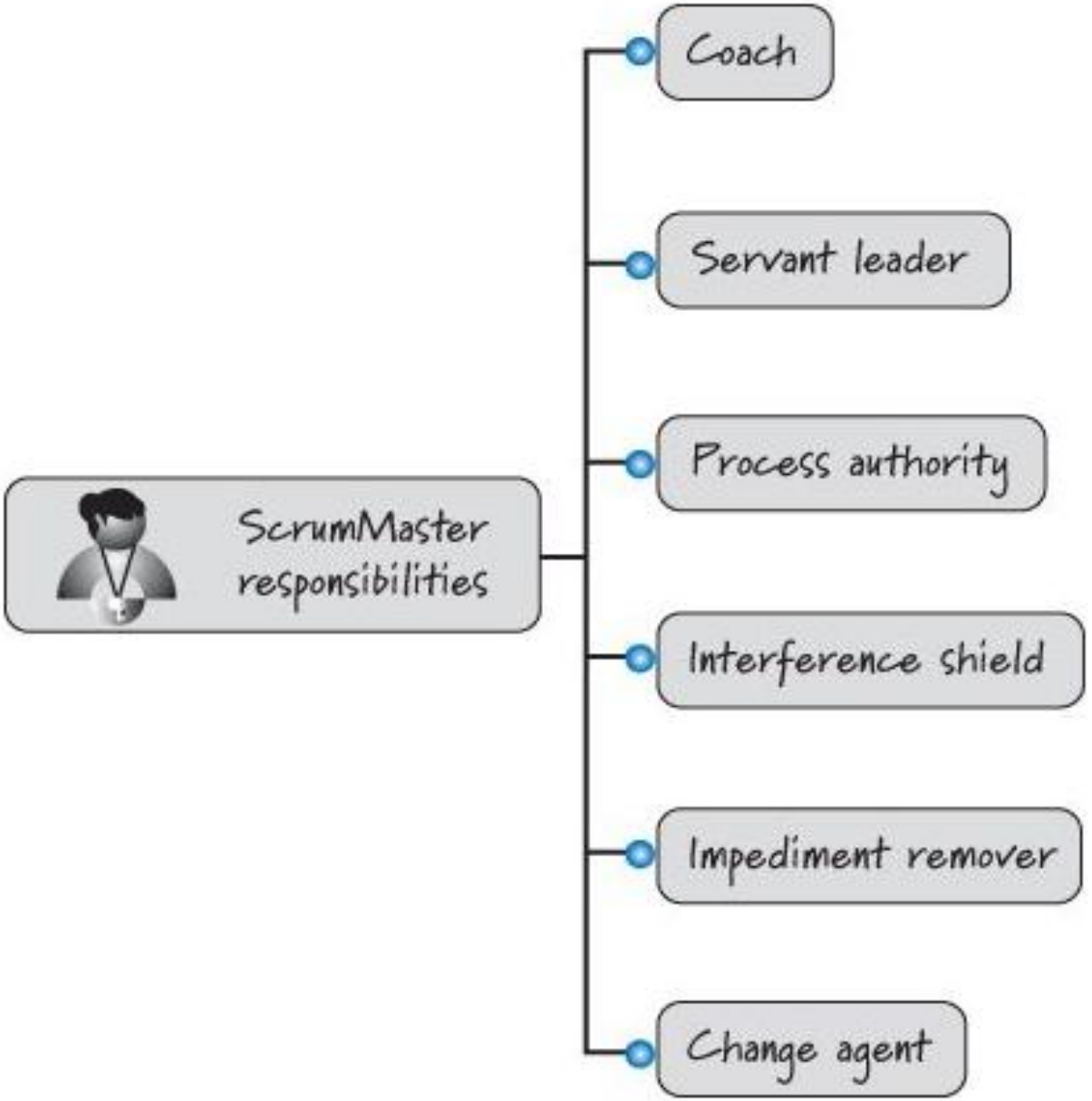


Introduction

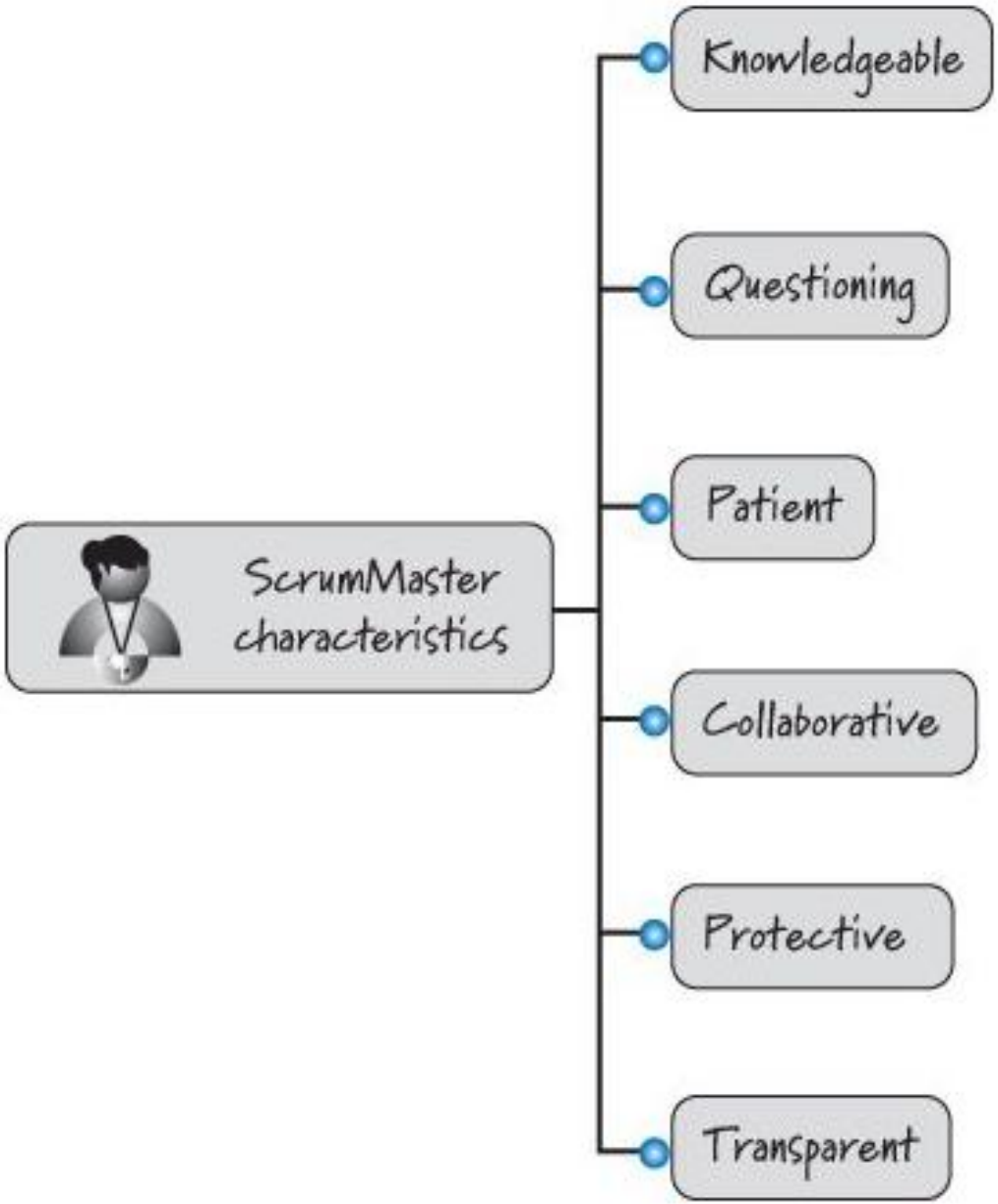
Introduction



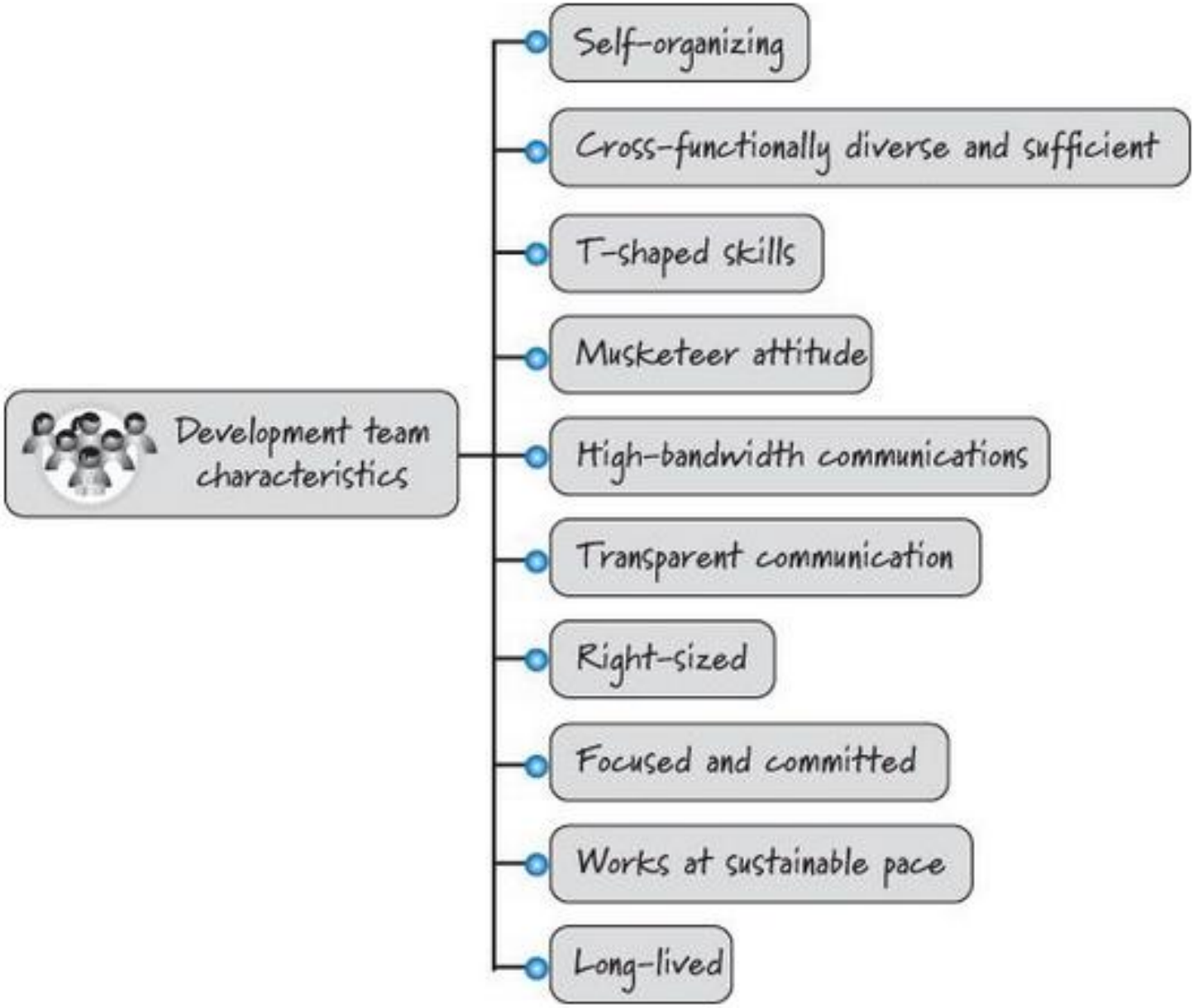
Introduction



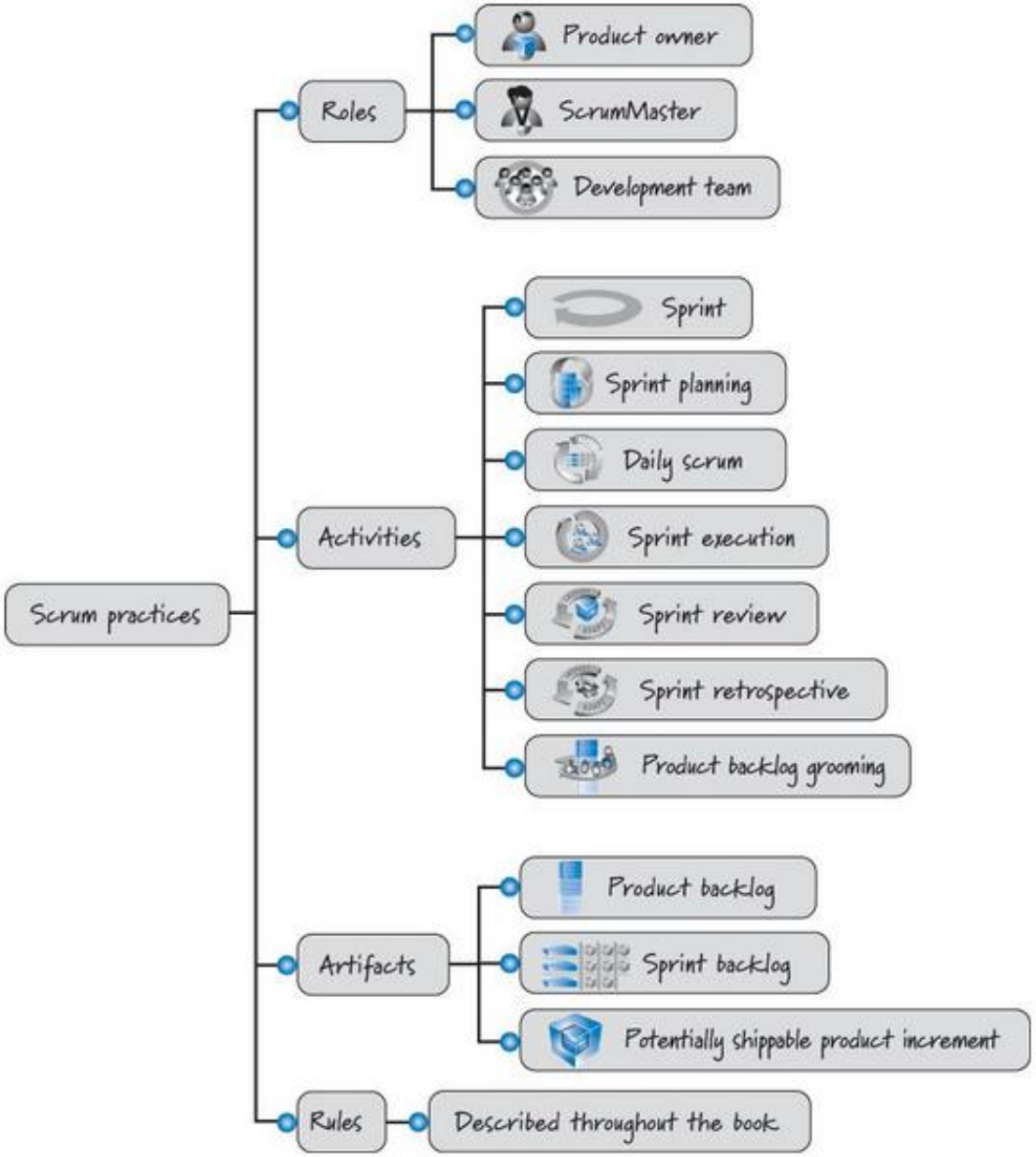
Introduction



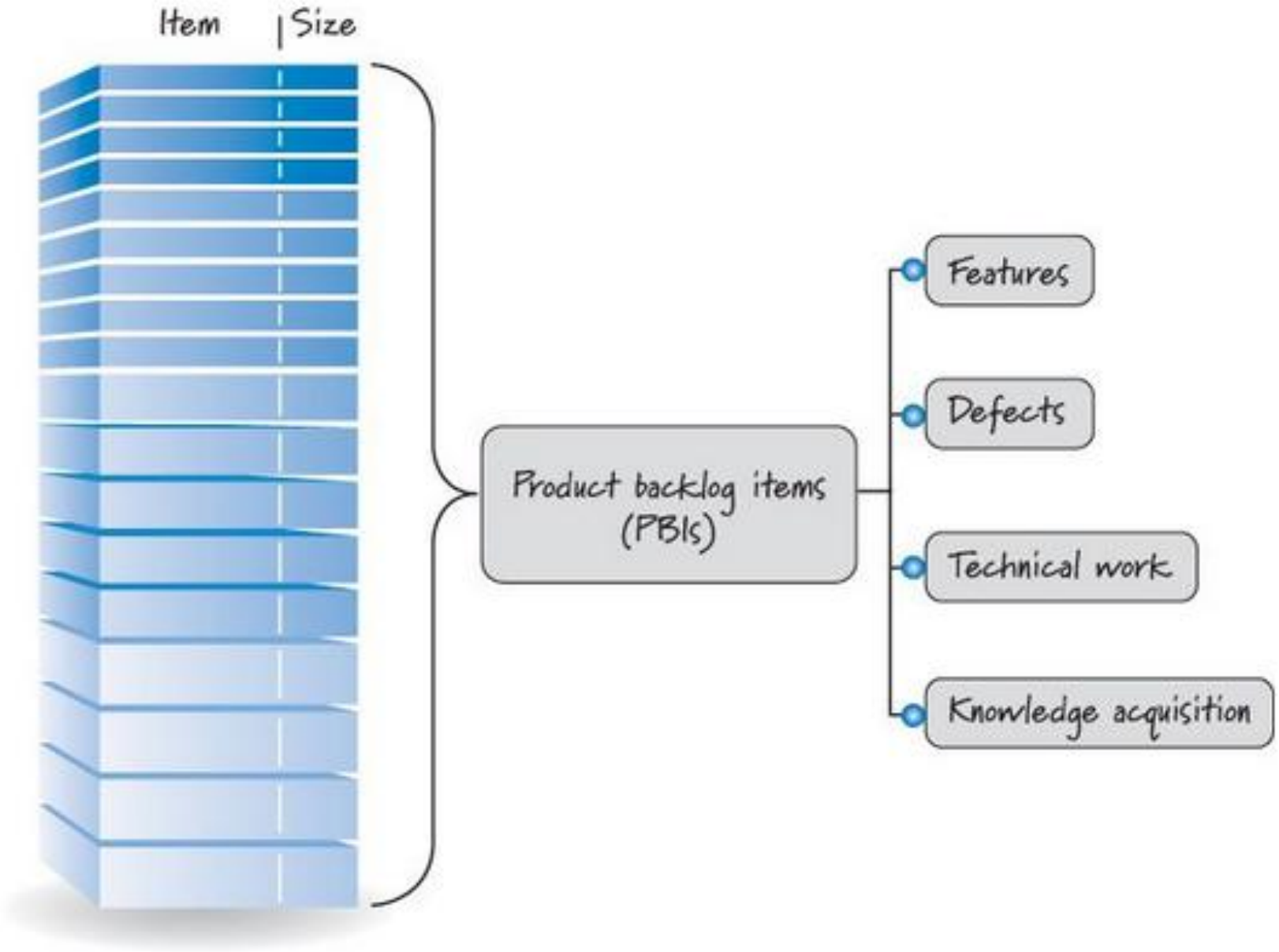
Introduction



Introduction



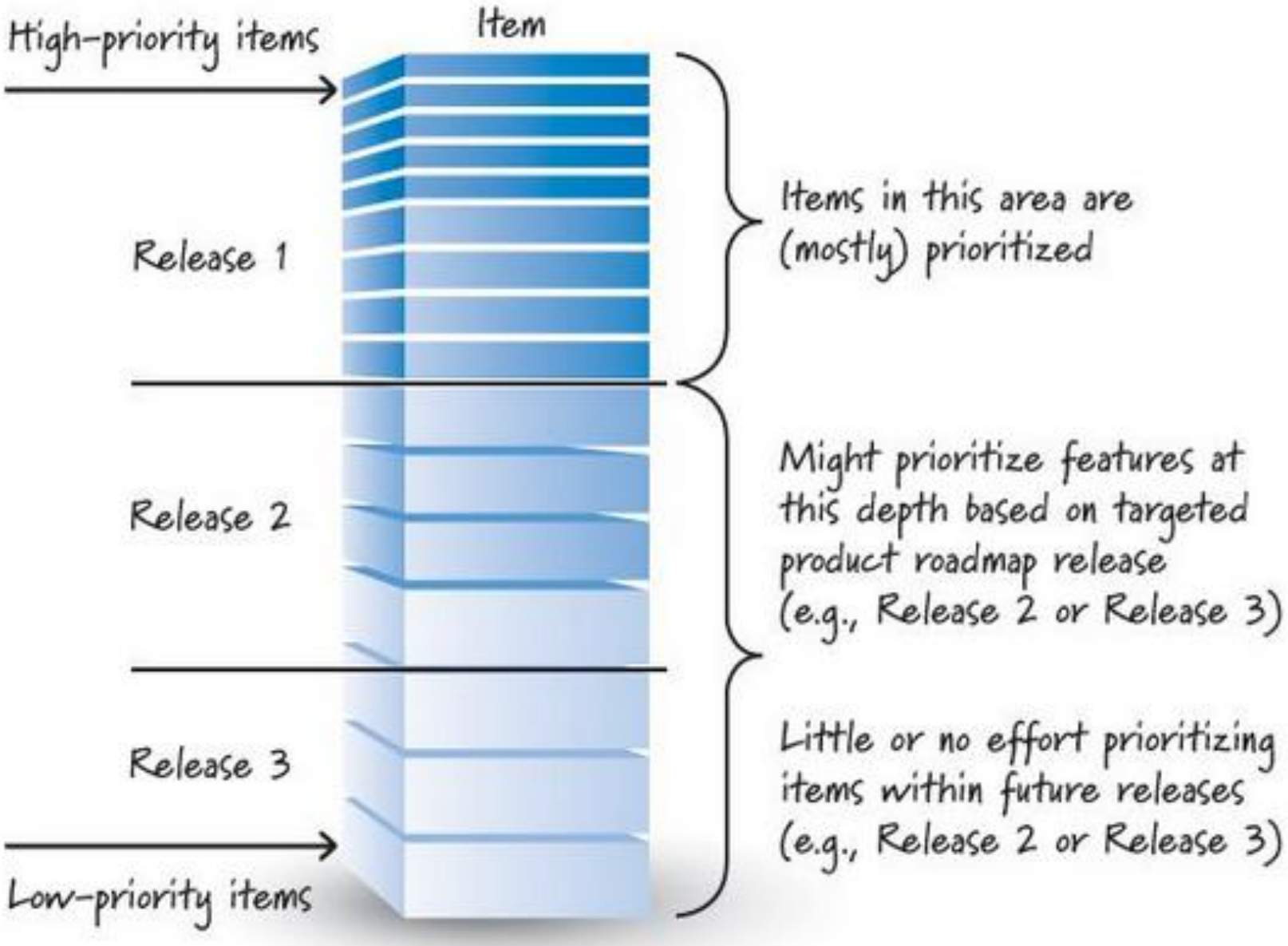
Introduction



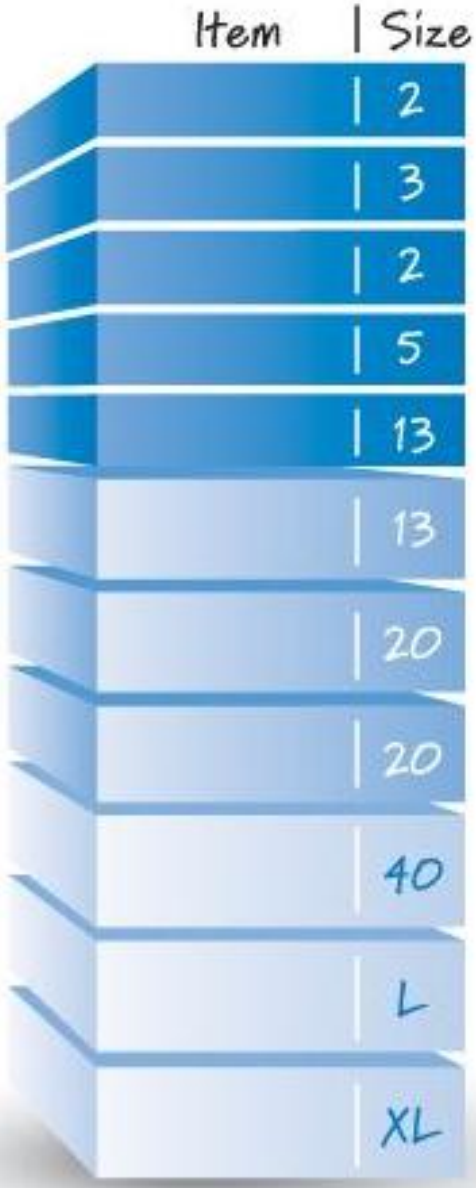
Introduction

PBI Type	Example
Feature	As a customer service representative I want to create a ticket for a customer support issue so that I can record and manage a customer's request for support.
Change	As a customer service representative I want the default ordering of search results to be by last name instead of ticket number so that it's easier to find a support ticket.
Defect	Fix defect #256 in the defect-tracking system so that special characters in search terms won't make customer searches crash.
Technical improvement	Move to the latest version of the Oracle DBMS.
Knowledge acquisition	Create a prototype or proof of concept of two architectures and run three tests to determine which would be a better approach for our product.

Introduction



Introduction

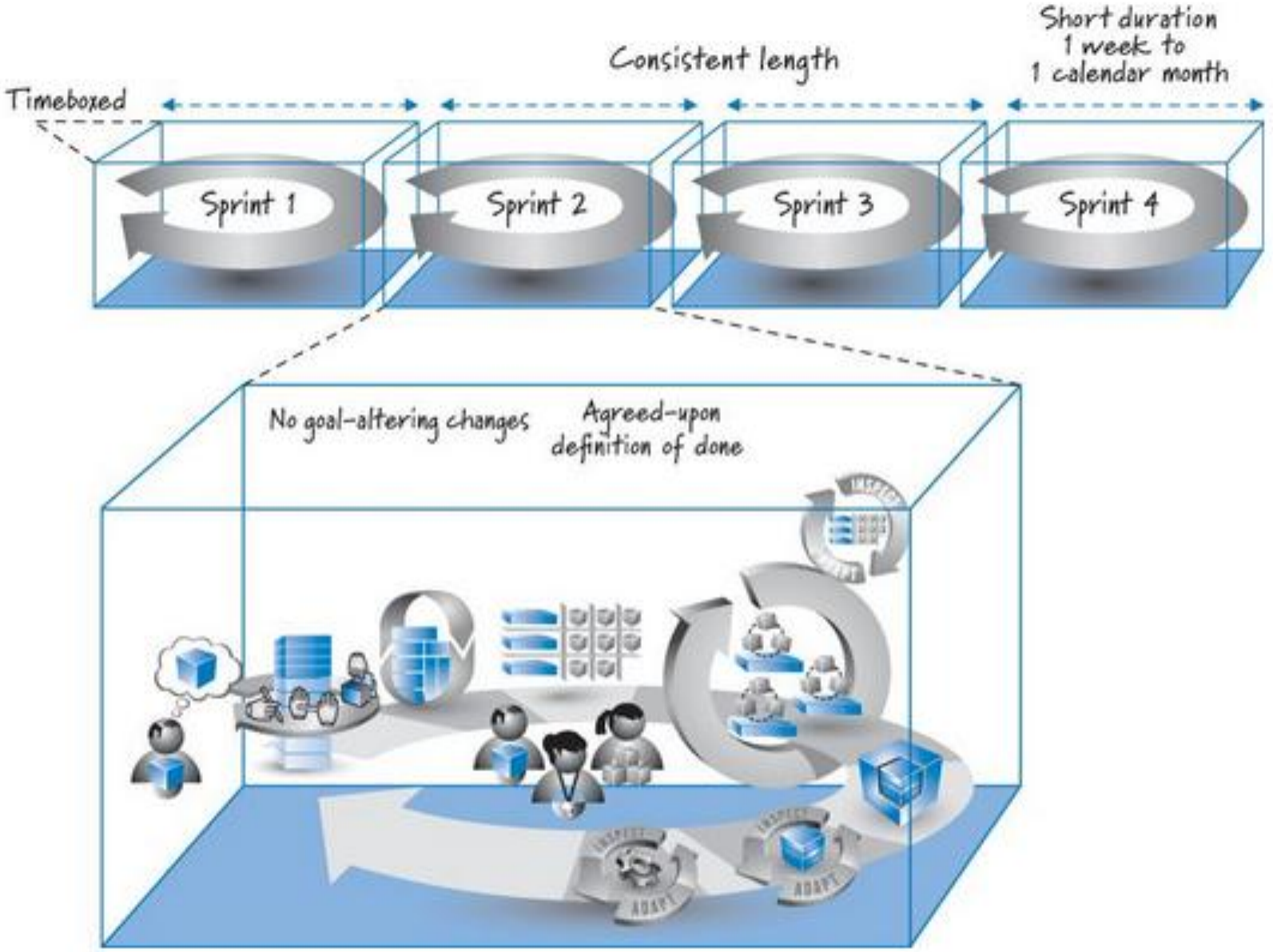


Each item has a size estimate

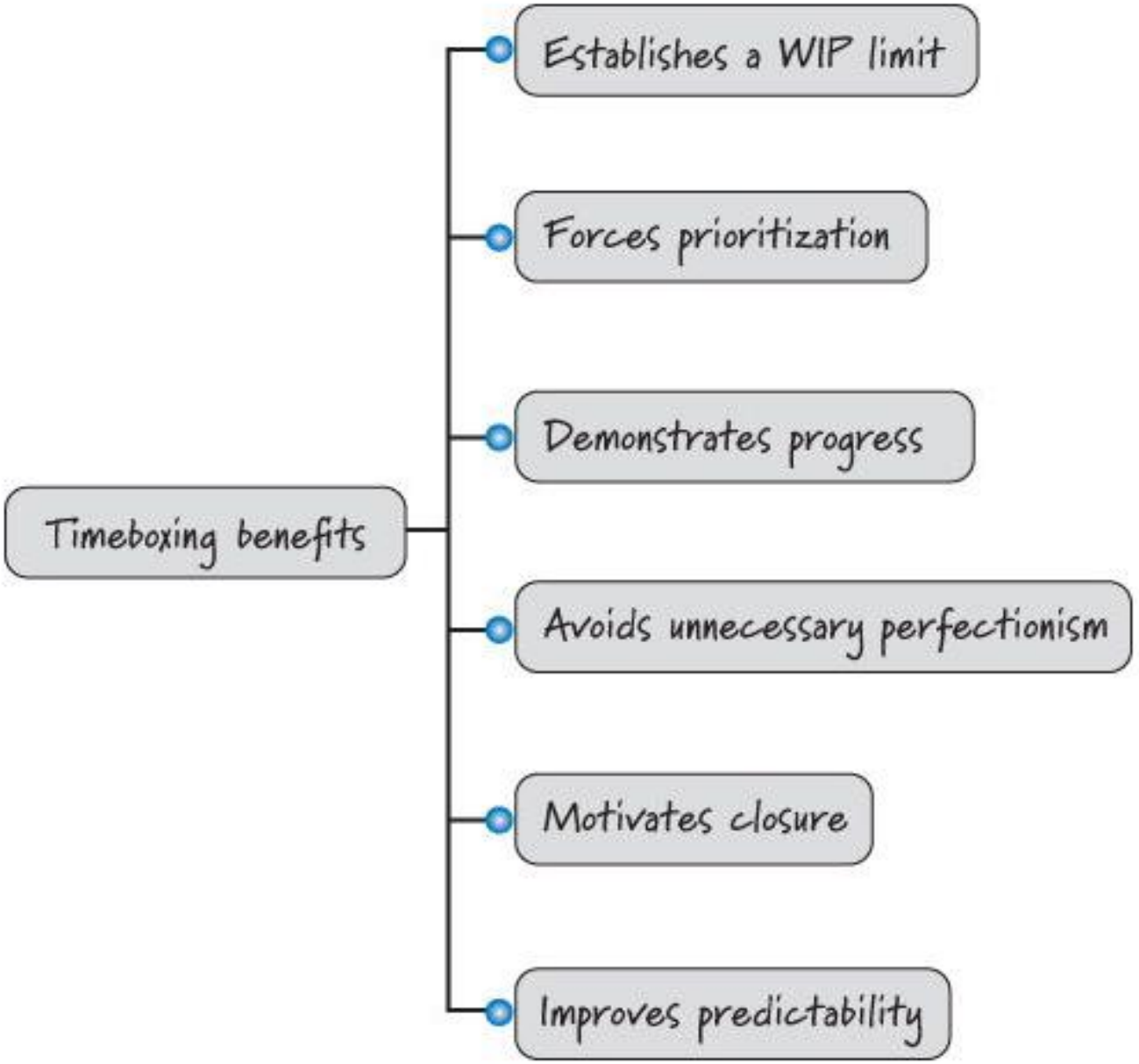
Most estimates are story point or ideal day estimates

Very large items near the bottom may not have an estimate or may be estimated in T-shirt sizes

Introduction



Introduction



Introduction

Scrum Events

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

Scrum Planning Meeting

The work to be performed in the Sprint is planned at the Sprint Planning Meeting. This plan is created by the collaborative work of the entire Scrum Team.

The Sprint Planning Meeting is time-boxed to eight hours for a one-month Sprint.

Daily Scrum Meeting

The Daily Scrum is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours. This is done by inspecting the work since the last Daily Scrum and forecasting the work that could be done before the next one.

Sprint Review

A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed. During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint.

Sprint Retrospective

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning Meeting. This is a three-hour time-boxed meeting for one-month Sprints. Proportionately less time is allocated for shorter Sprints.

Introduction

Purpose of Sprint Retrospective

- Inspect how the last sprint went with regards to people, process, relationships and tools.
- Identify and order the major items that went well and potential improvements.
- Creating plan for implementing improvements for the sprints onwards.

Introduction

What Are User Stories?

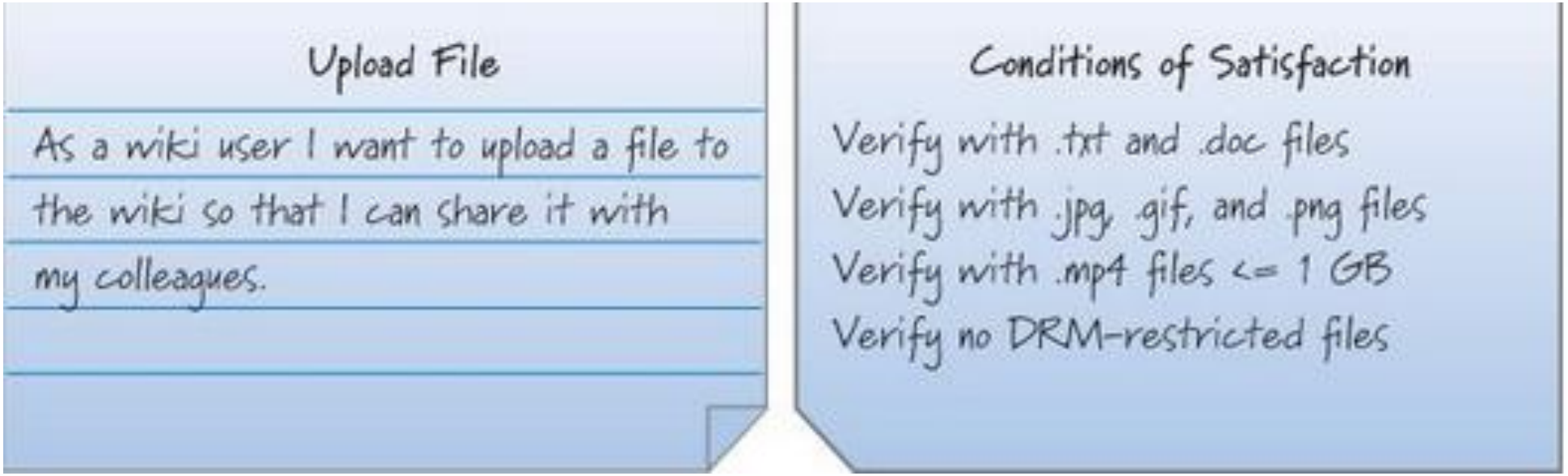
User stories are a convenient format for expressing the desired business value for many types of product backlog items, especially features. User stories are crafted in a way that makes them understandable to both business people and technical people. They are structurally simple and provide a great placeholder for a conversation. Additionally, they can be written at various levels of granularity and are easy to progressively refine.

As well adapted to our needs as user stories might be, I don't consider them to be the only way to represent product backlog items. They are simply a lightweight approach that dovetails nicely with core agile principles and our need for an efficient and effective placeholder. I use them as the central placeholder to which I will attach any other information that I think is relevant and helpful for detailing a requirement. If I find that user stories are a forced fit for a particular situation (such as representing certain defects), I'll use another approach. For example, I once saw a team write the following user story: "As a customer I would like the system to not corrupt the database." I think we can all agree that a user story is probably not the best way to represent this issue. Perhaps a simple reference to the defect in the defect-tracking system would be more appropriate.

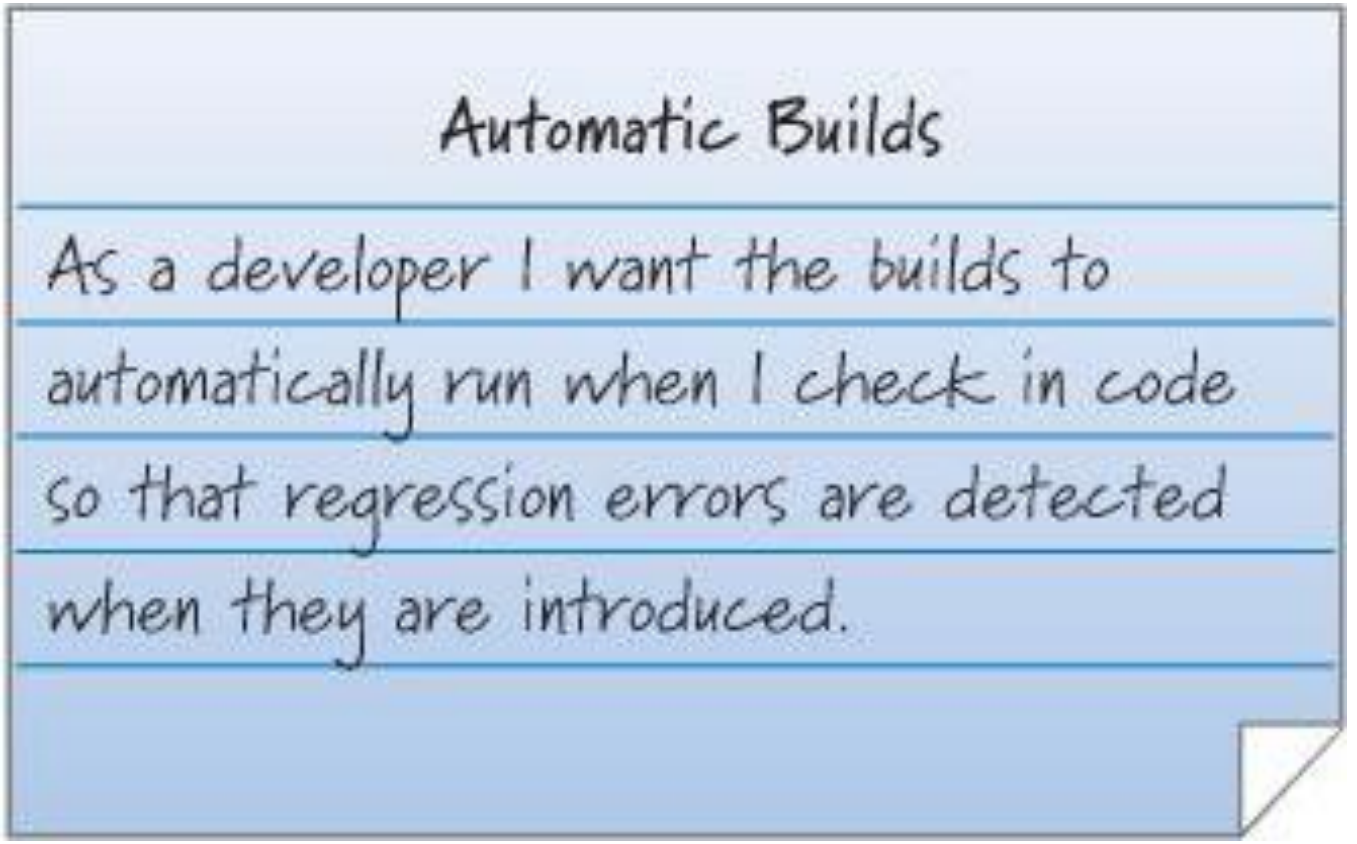
So what exactly are user stories? Ron Jeffries offers a simple yet effective way to think about user stories (Jeffries 2001). He describes them as the three Cs: card, conversation, and confirmation.

Introduction

Introduction



Introduction

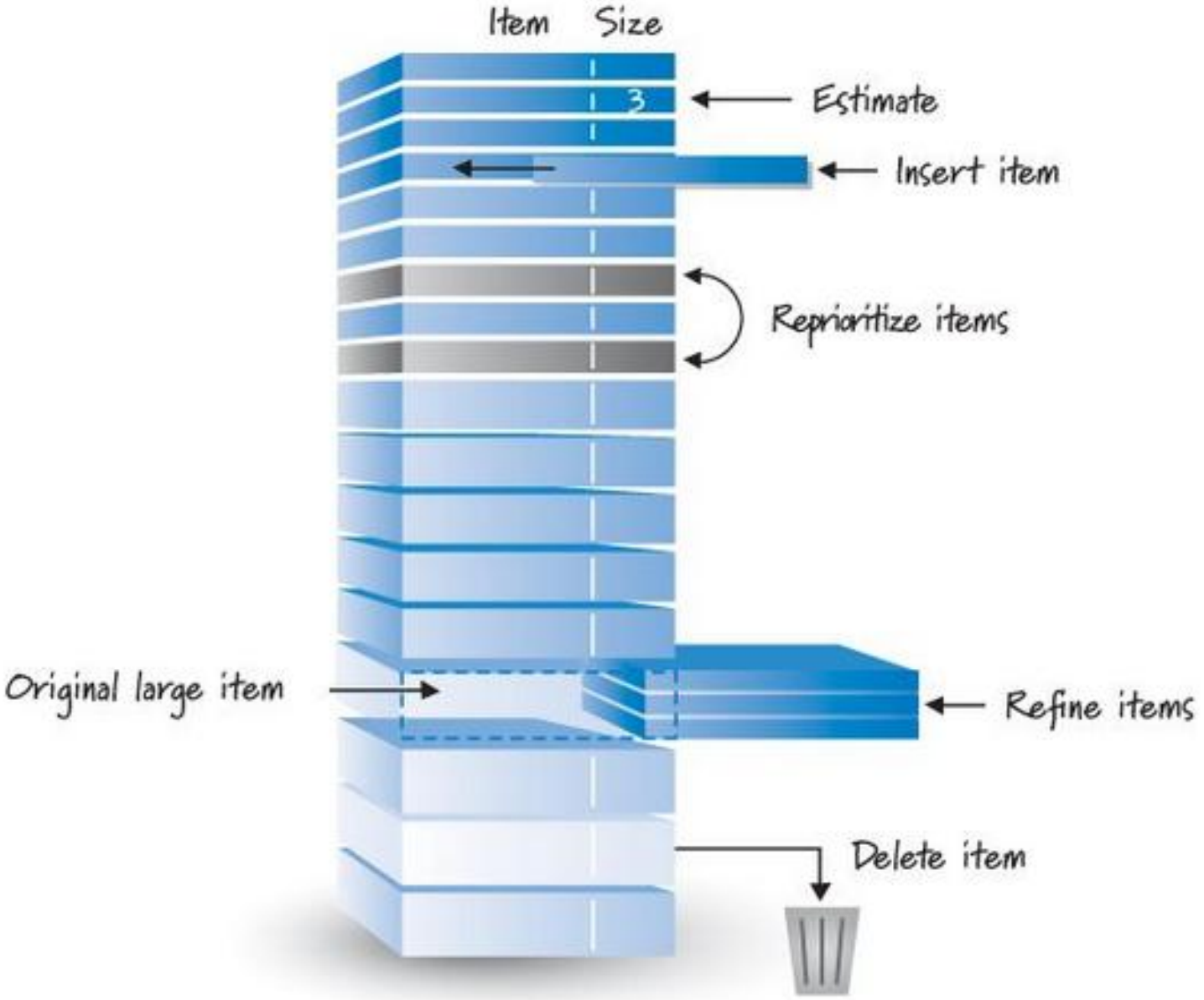


Introduction

What Is Grooming?

Grooming refers to a set of three principal activities: creating and refining (adding details to) PBIs, estimating PBIs, and prioritizing PBIs.

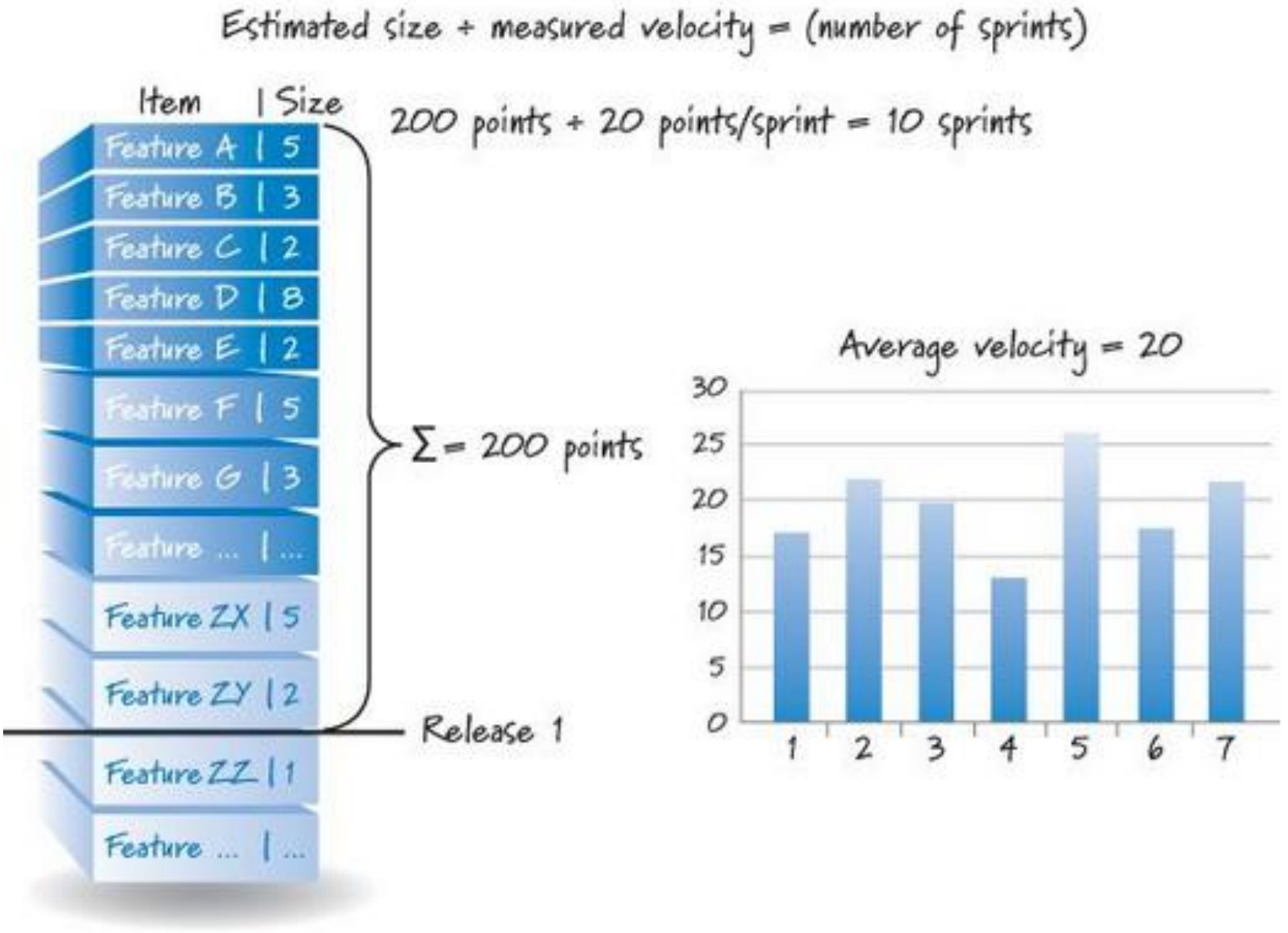
Introduction



Introduction

	Definition of Done
❑	Design reviewed
❑	Code completed
❑	Code refactored
❑	Code in standard format
❑	Code is commented
❑	Code checked in
❑	Code inspected
❑	End-user documentation updated
❑	Tested
❑	Unit tested
❑	Integration tested
❑	Regression tested
❑	Platform tested
❑	Language tested
❑	Zero known defects
❑	Acceptance tested
❑	Live on production servers

Introduction



Introduction