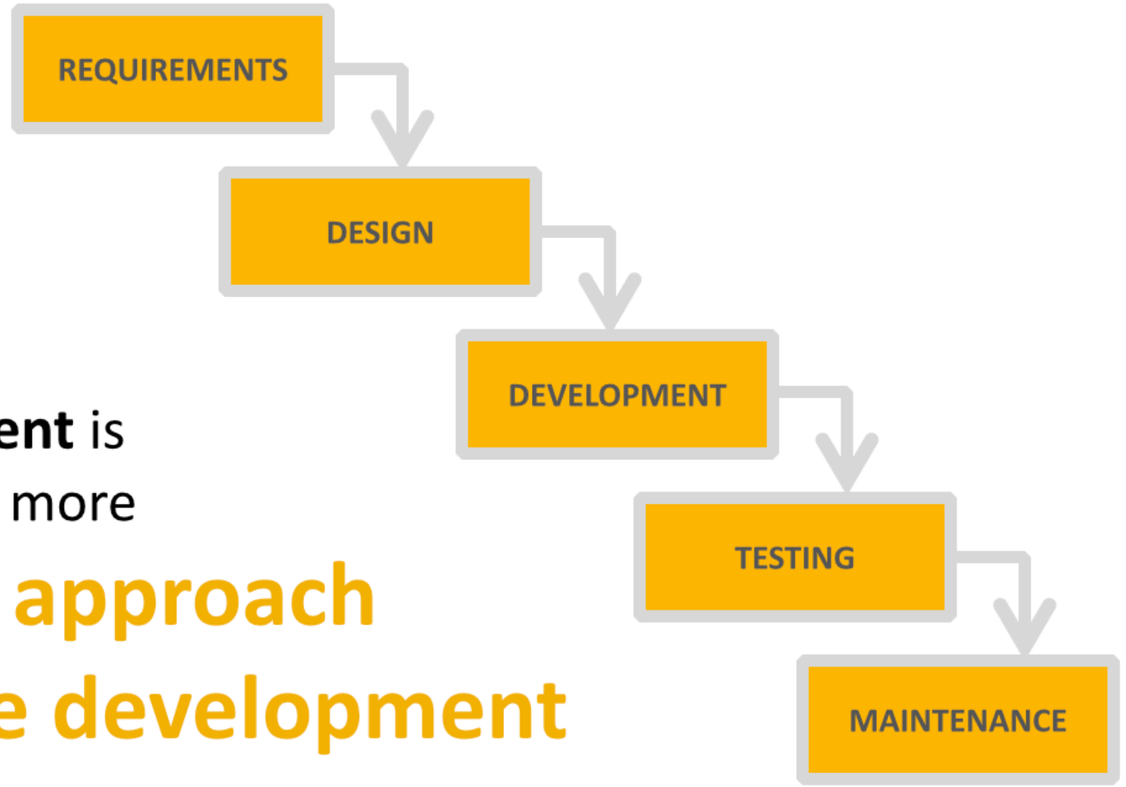
AGILE

INTRODUCTION

LIFE BEFORE AGILE



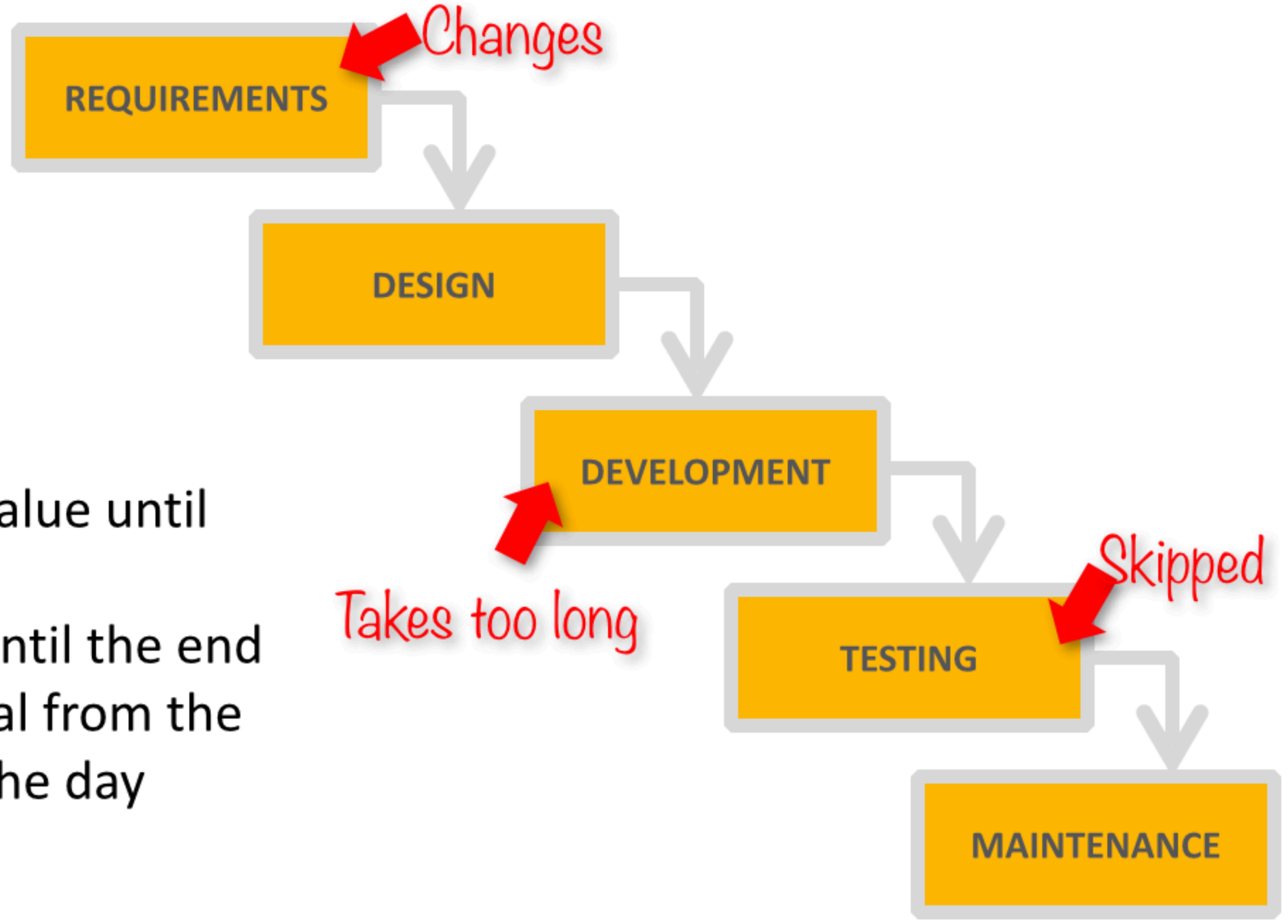
Waterfall Development is another name for the more

traditional approach to software development

WATERFALL DEVELOPMENT

You **complete one phase** (e.g. design) **before** moving on to the **next phase** (e.g. development)

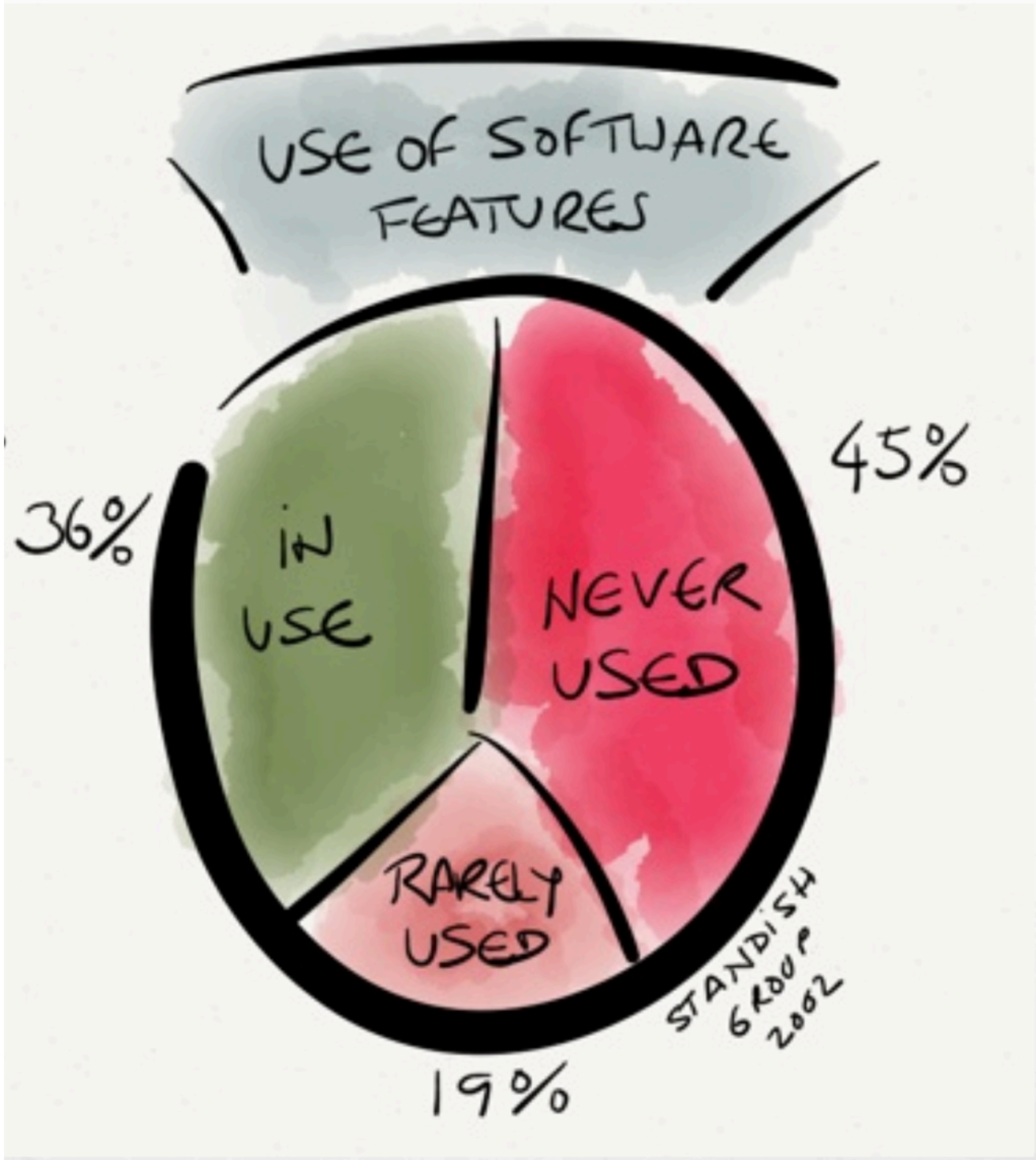
You **rarely aim to re-visit a 'phase' once it's completed**. That means, you **better get whatever you're doing right the first time!**



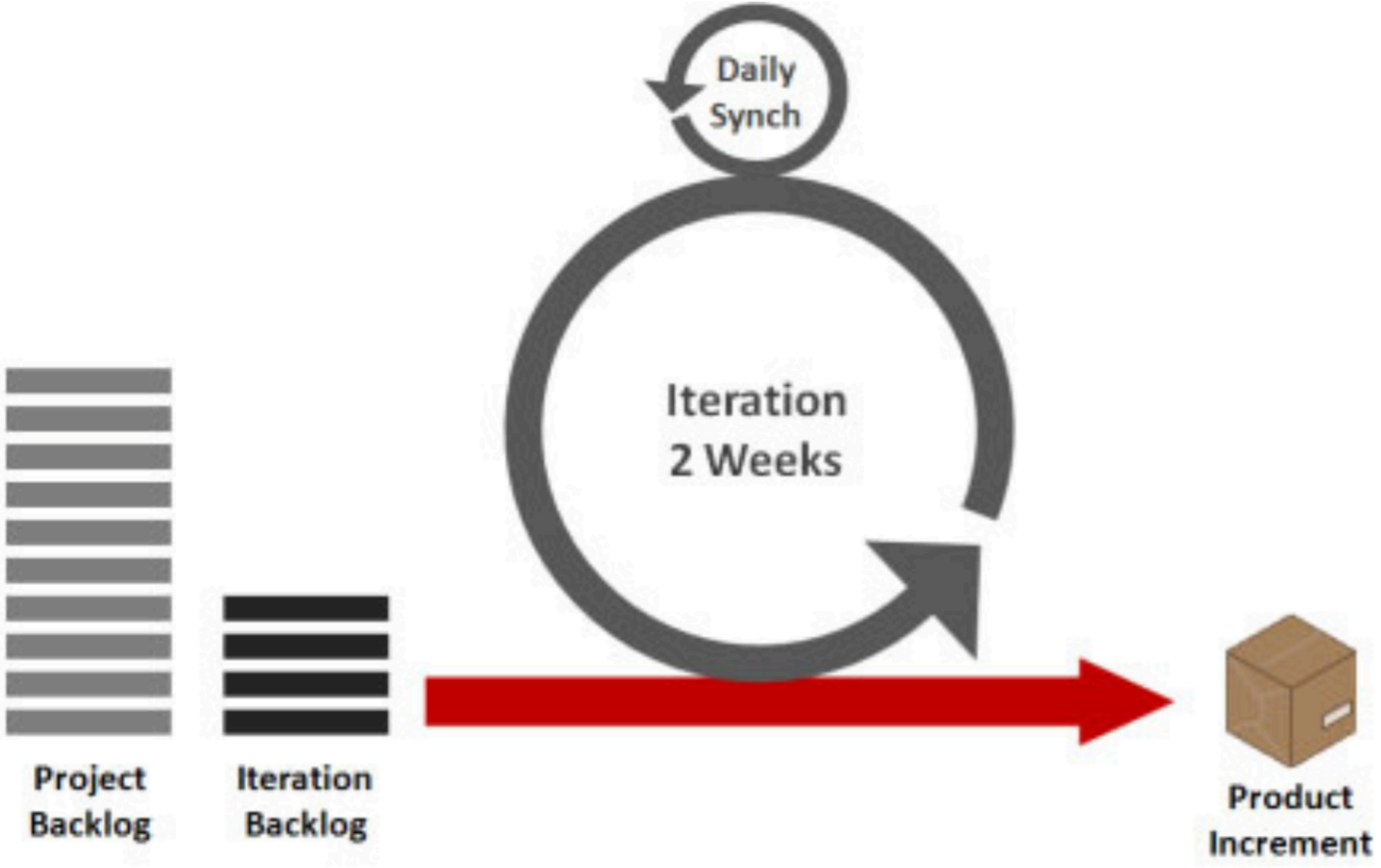
- You don't realize any value until the end of the project
- You leave the testing until the end
- You don't seek approval from the stakeholders until late in the day

This approach is **highly risky**, often more **costly** and

WASTEFUL PRODUCTS



Not a process, it's a **philosophy** or **set of values**


Daily
Synch
Iteration
2 Weeks
Project
Backlog
Iteration
Backlog
Product
Increment

AGILE MANIFESTO

INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS
WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION
RESPONDING TO CHANGE OVER FOLLOWING A PLAN

AGILE PRINCIPLES

» Customer satisfaction by rapid. continuous delivery of useful software

» Working software is delivered frequently (weeks rather than months)

» Working software is the principal measure of progress

» Even late changes in requirements are welcomed

» Close, daily cooperation between business people and developers

AGILE PRINCIPLES (CONTINUED)

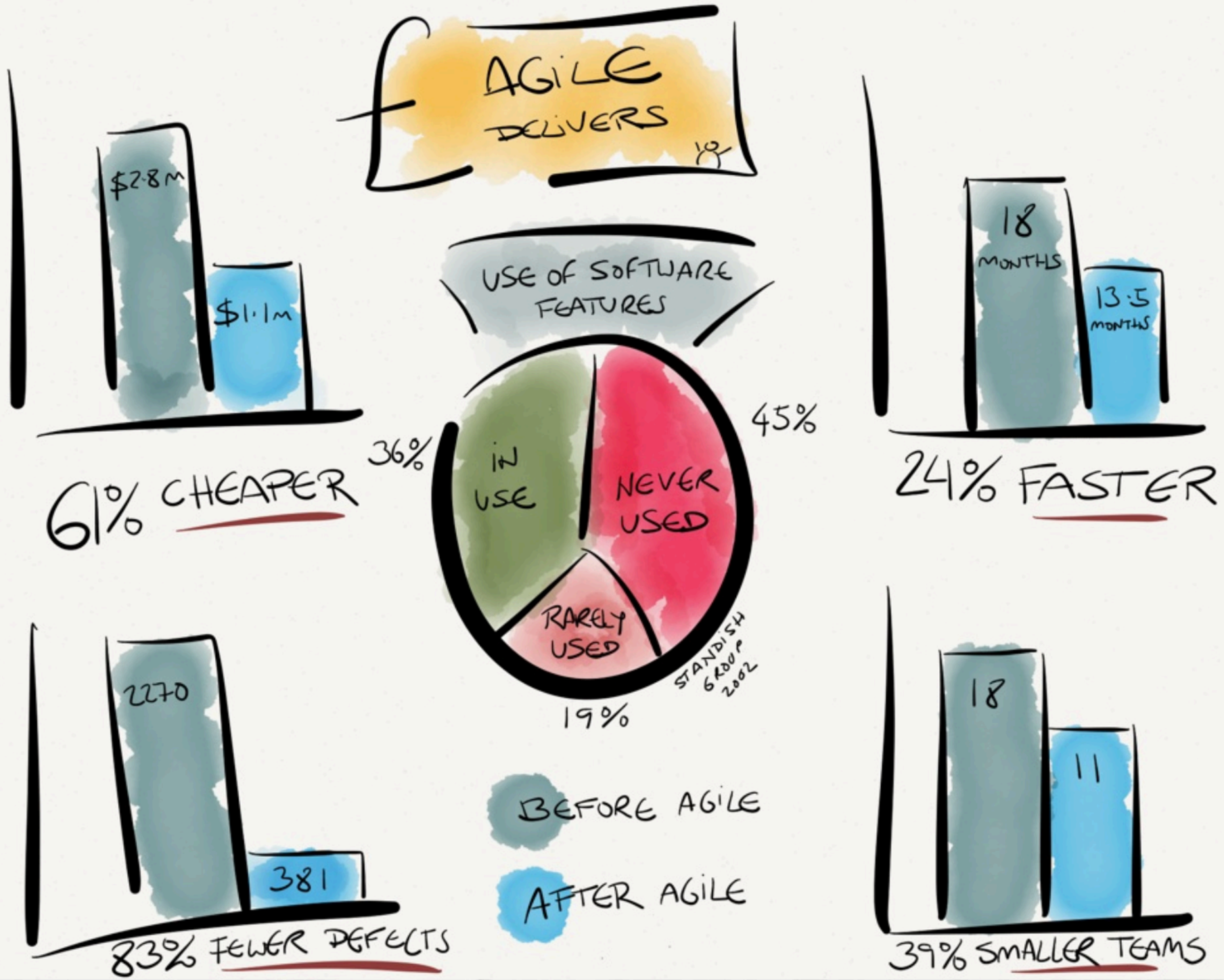
» Projects are built around motivated individuals, who should be trusted

» Continuous attention to technical excellence and good design

» Simplicity

» Self-organizing teams

» Regular adaptation to changing circumstances


AGILE
DELIVERS
$2.8M
$1.1M
61% CHEAPER
USE OF SOFTWARE FEATURES
36%
IN USE
45%
NEVER USED
RARELY USED
19%
STANDISH GROUP 2002
18
MONTHS
13.5
MONTHS
24% FASTER
2270
381
83% FEWER DEFECTS
BEFORE AGILE
AFTER AGILE
18
11
39% SMALLER TEAMS

SOME NUMBERS TO CONSIDER

» 93% increased productivity

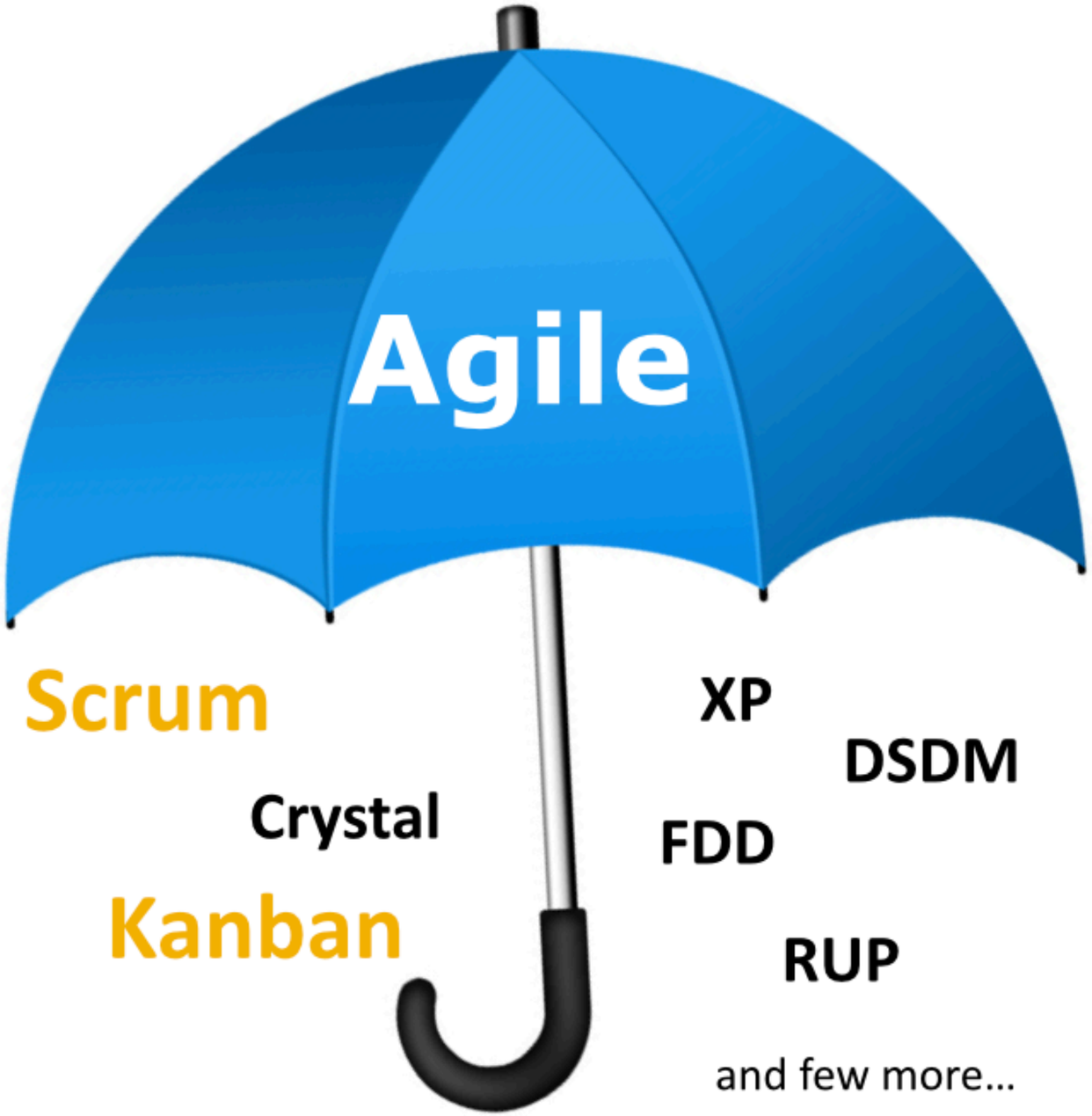
» 88% increased quality

» 83% improved stakeholder satisfaction

» 49% reduced costs

» 66% three-year, risk adjusted return on investment

AGILE PROCESSES & TOOLS

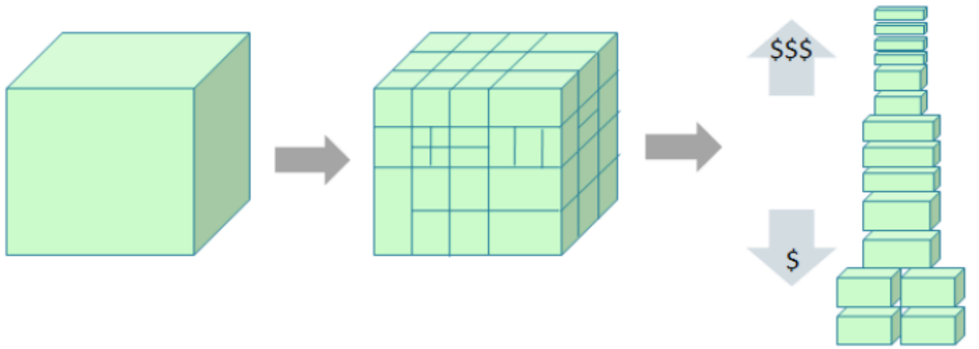


A **light-weight agile process tool**

Scrum

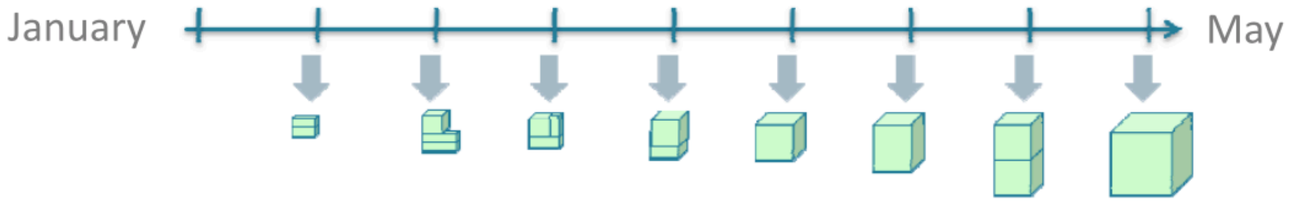
Split your organization
into small, cross-functional, self-organizing teams.

Split your work into a list of small, concrete deliverables. Sort the list by priority and estimate the relative effort of each item.



SCRUM PROCESS

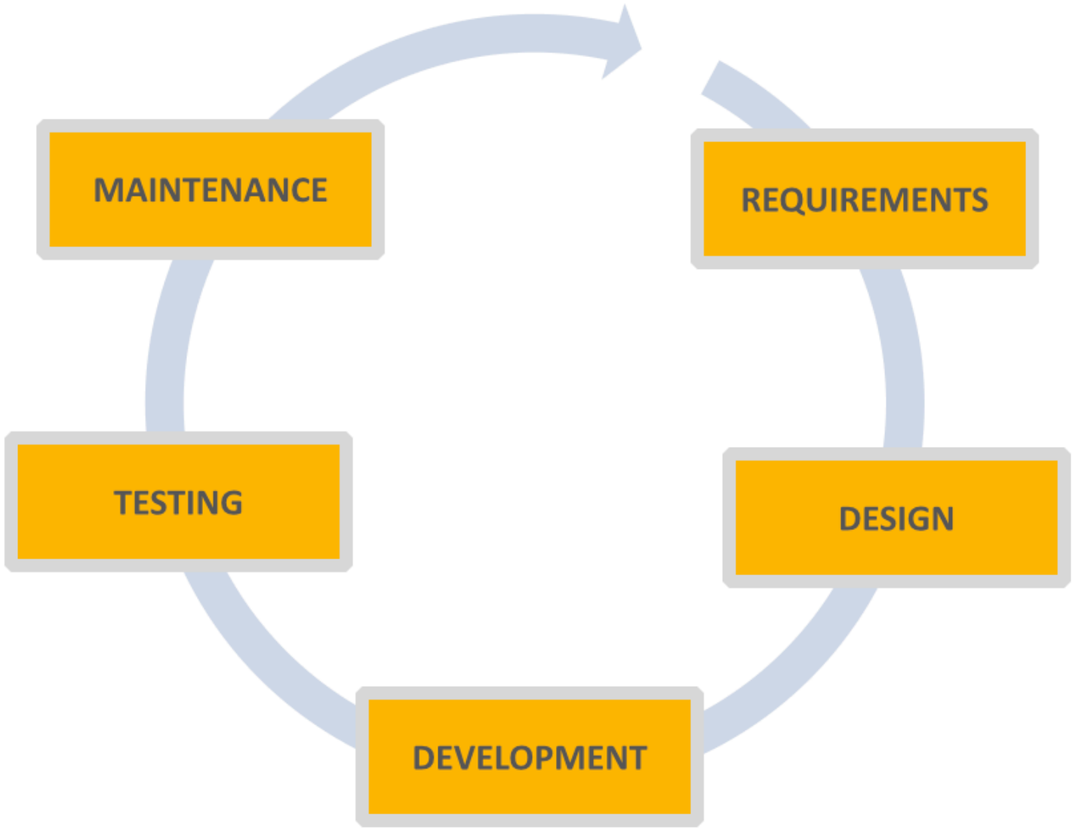
Split time into short fixed-length iterations/ sprints (usually 2 – 4 weeks), with potentially shippable code demonstrated after each iteration.



Optimize the release plan and update priorities in collaboration with the customer, based on insights gained by inspecting the release after each iteration.

Optimize the process by having a retrospective after each iteration.

SCRUM VS. WATERFALL



THINGS WE DO IN SCRUM

The project/ product is described as a list of features: the **backlog**.

The features are described in terms of **user stories**.

The scrum team **estimates** the **work** associated with each story.

Features in the backlog are **ranked** in order of importance.

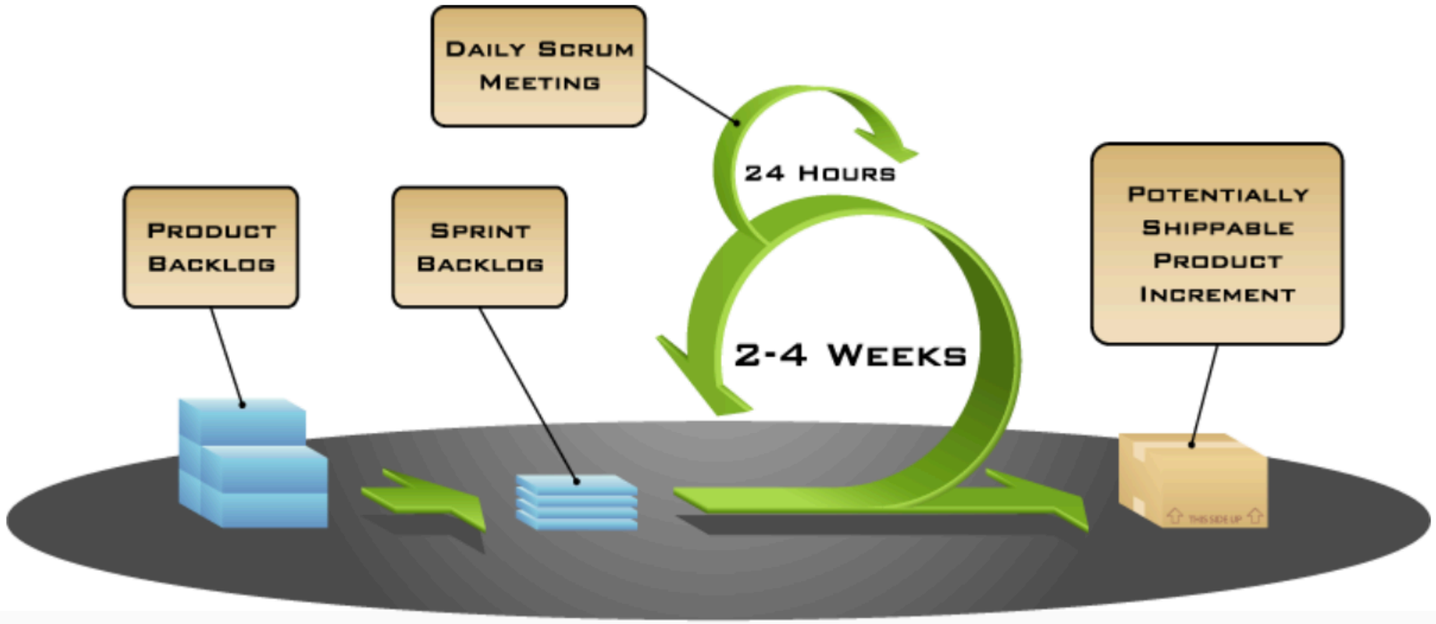
Result: a **ranked** and **weighted** list of product features, a **roadmap**.

Daily scrum meeting to discuss **What did you do y'day**? **What will you do today**? **Any obstacles**?

SCRUM IN A NUTSHELL

So instead of a **large group** spending **a long time** building a **big thing**, we have a **small team** spending a **short time** building a **small thing**.

But integrating regularly to see the whole.



KANBAN

LEAN APPROACH TO AGILE DEVELOPMENT

AIM IS TO ELIMINATE ‘WASTE’ WHEREVER POSSIBLE...

Limit Work-In-Progress
Visualize the
Visual Card
KANBAN
Work
Signboard
Measure & Manage Flow
Just-in-time (JIT)

LEAN PRINCIPLES

» Continuous improvement

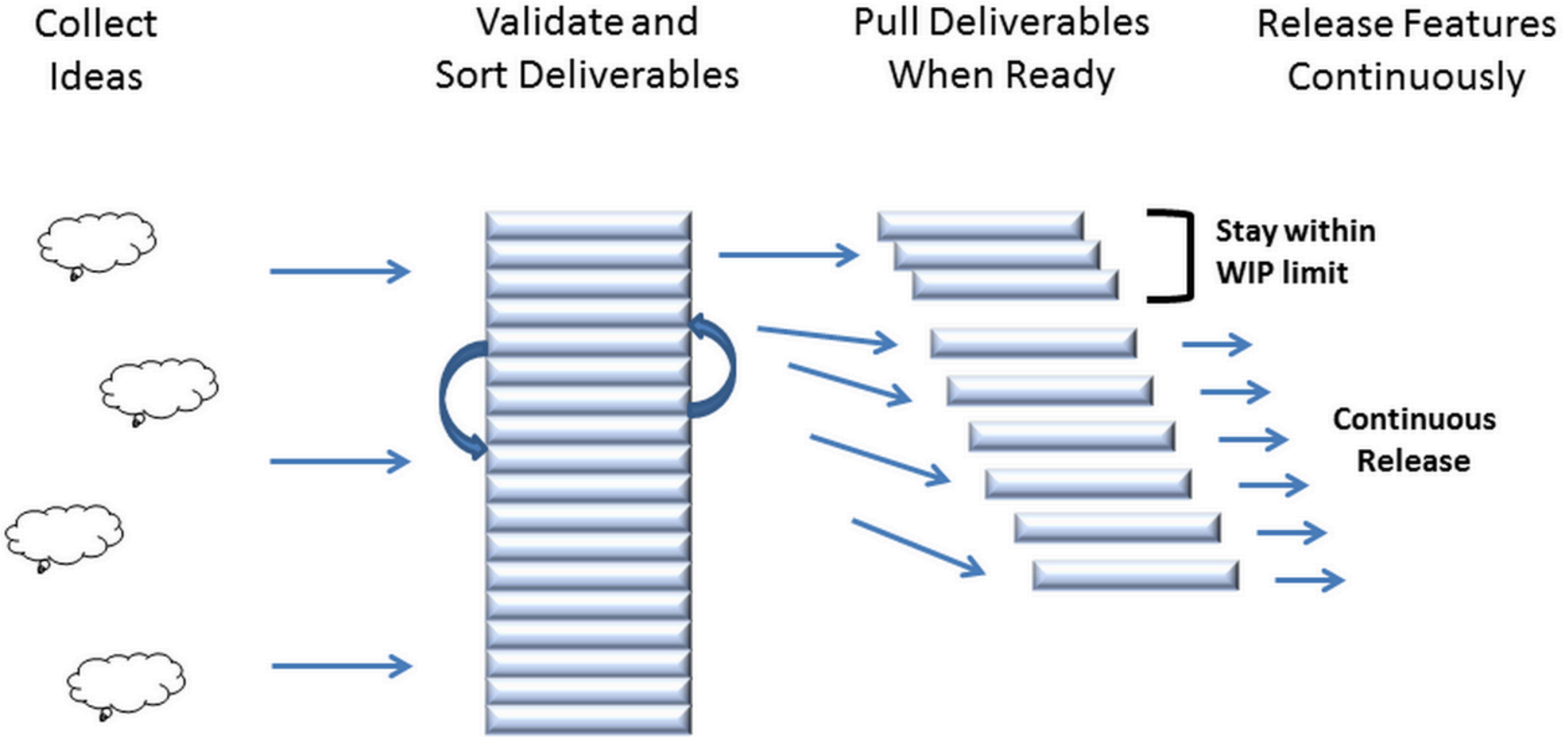
» Make batches of work smaller ("one-piece flow")

» Use the principle of pull ("pull what is ready")

» Limit "Work in Progress"

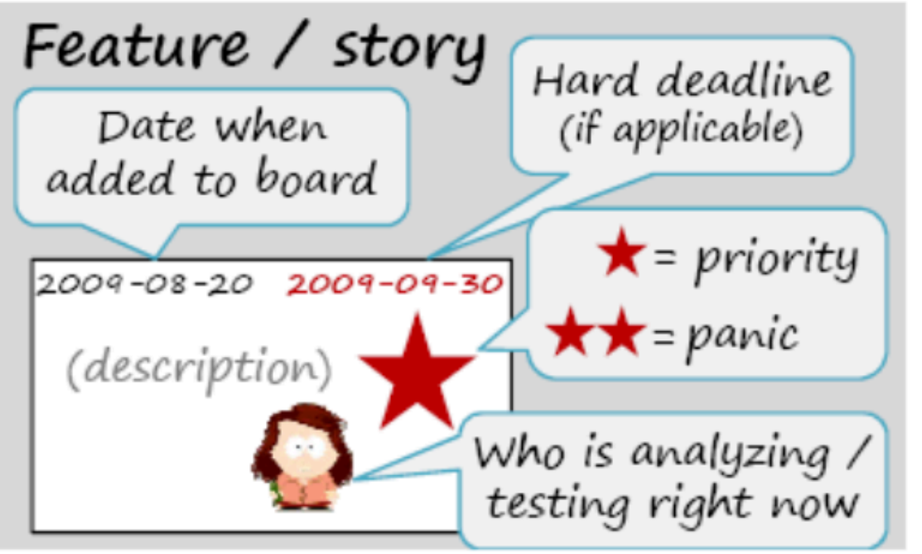
» Automate everything

» Release when ready


Collect Ideas
Validate and Sort Deliverables
Pull Deliverables When Ready
Release Features Continuously
Stay within WIP limit
Continuous Release

Visualize the workflow

- Split the work into pieces, write each item on a card and put on the wall
- Use named columns to illustrate where each item is in the workflow

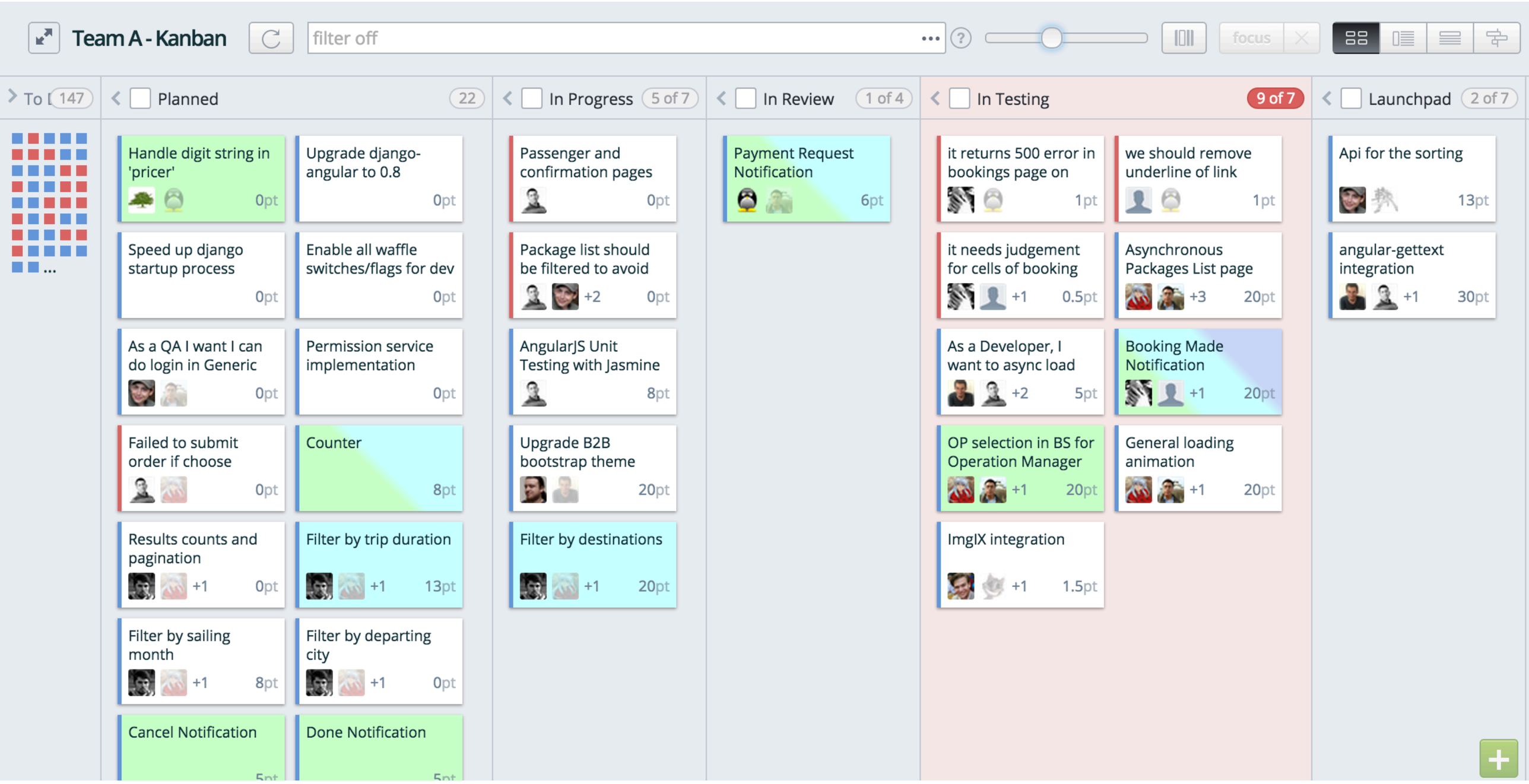


Limit WIP (work in progress)

- Assign explicit limits to how many items may be in progress at each stage

Measure the lead time (average time to complete one item, sometimes called “cycle time”)

- Optimize the process to make lead time as small and predictable as possible


Team A - Kanban
filter off
focus
To D 147
Planned 22
In Progress 5 of 7
In Review 1 of 4
In Testing 9 of 7
Launchpad 2 of 7
Handle digit string in 'pricer' 0pt
Upgrade django-angular to 0.8 0pt
Speed up django startup process 0pt
Enable all waffle switches/flags for dev 0pt
As a QA I want I can do login in Generic 0pt
Permission service implementation 0pt
Failed to submit order if choose 0pt
Counter 8pt
Results counts and pagination +1 0pt
Filter by trip duration +1 13pt
Filter by sailing month +1 8pt
Filter by departing city +1 0pt
Cancel Notification 5pt
Done Notification 5pt
Passenger and confirmation pages 0pt
Package list should be filtered to avoid +2 0pt
AngularJS Unit Testing with Jasmine 8pt
Upgrade B2B bootstrap theme 20pt
Filter by destinations +1 20pt
Payment Request Notification 6pt
it returns 500 error in bookings page on 1pt
we should remove underline of link 1pt
it needs judgement for cells of booking +1 0.5pt
Asynchronous Packages List page +3 20pt
As a Developer, I want to async load +2 5pt
Booking Made Notification +1 20pt
OP selection in BS for Operation Manager +1 20pt
General loading animation +1 20pt
ImgIX integration +1 1.5pt
Api for the sorting 13pt
angular-gettext integration +1 30pt

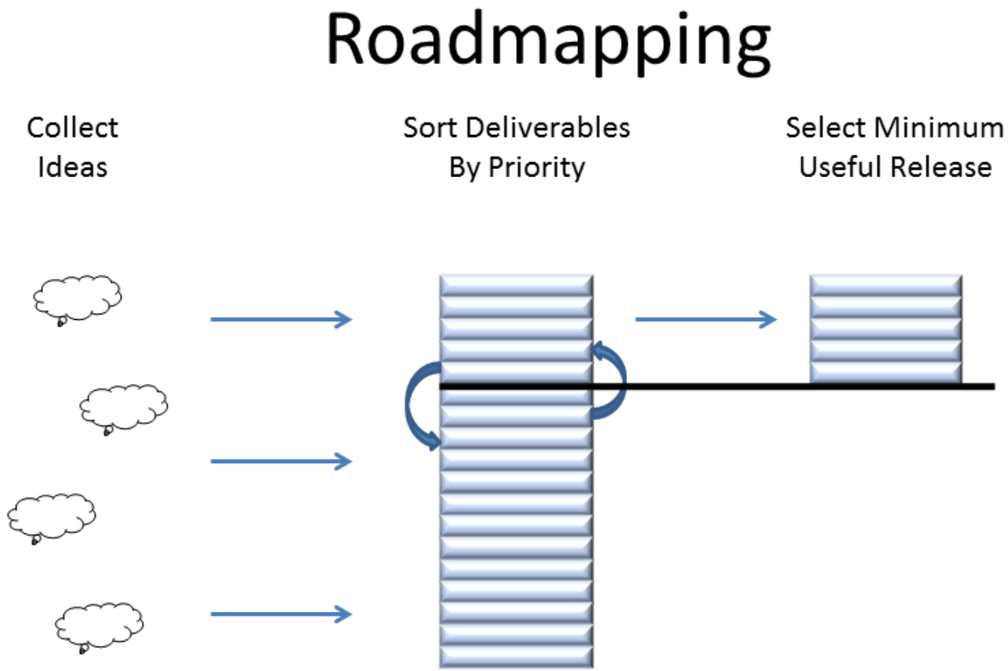
AGILE

@BESTOURS SOFTWARE

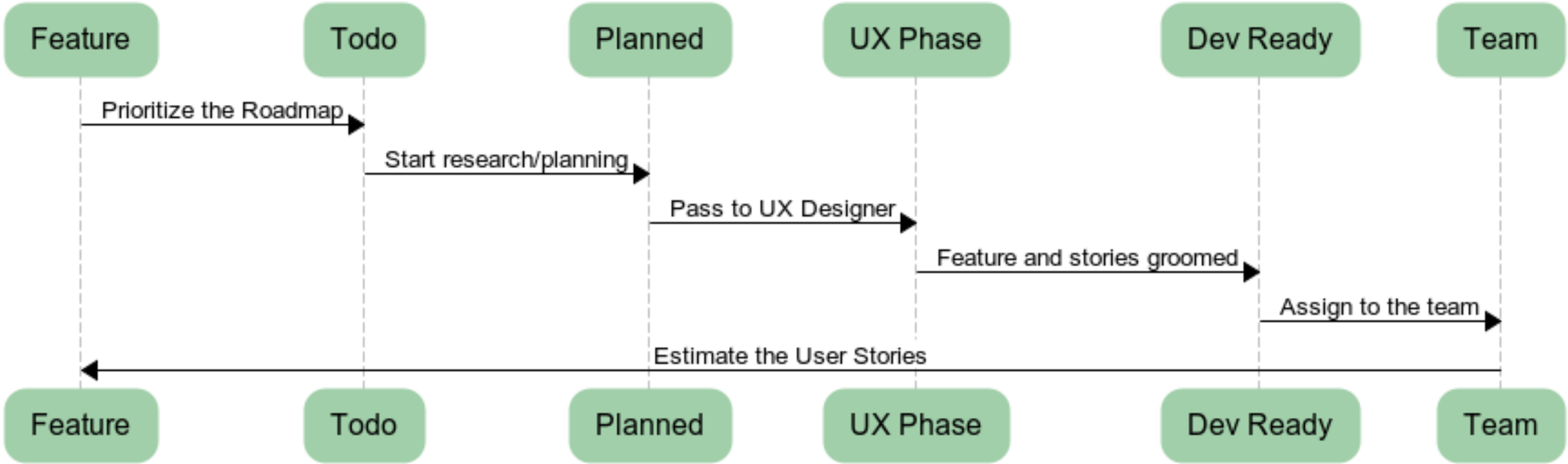
PRODUCT PLANNING

» Roadmapping

» Feature Prioritization



FEATURE PLANNING



FEATURE DEVELOPMENT

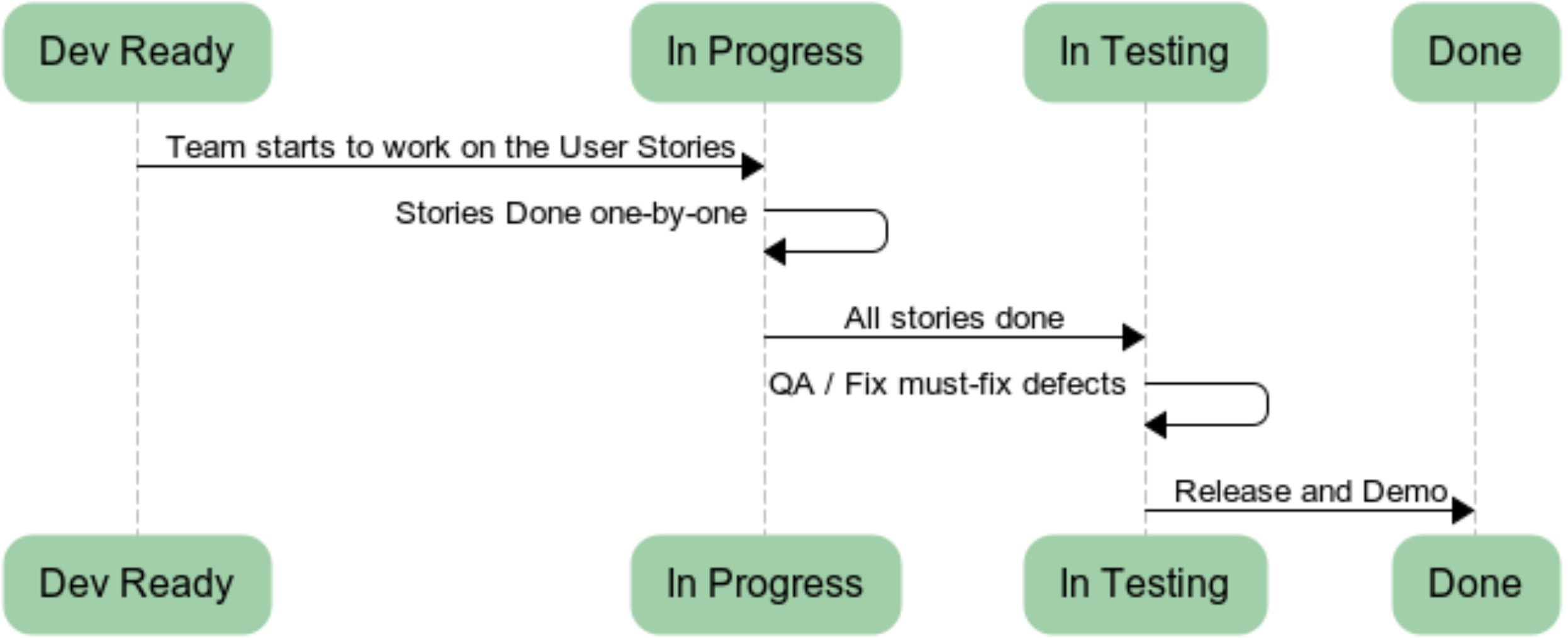
» Any Feature, Any Product, Any Team

» Continuous Delivery

» Pull requirements when ready

» Always work on the most valuable items

» Eliminate bottlenecks


Dev Ready
In Progress
In Testing
Done
Team starts to work on the User Stories
Stories Done one-by-one
All stories done
QA / Fix must-fix defects
Release and Demo
Dev Ready
In Progress
In Testing
Done

SCRUMBAN

» Lean approach to development

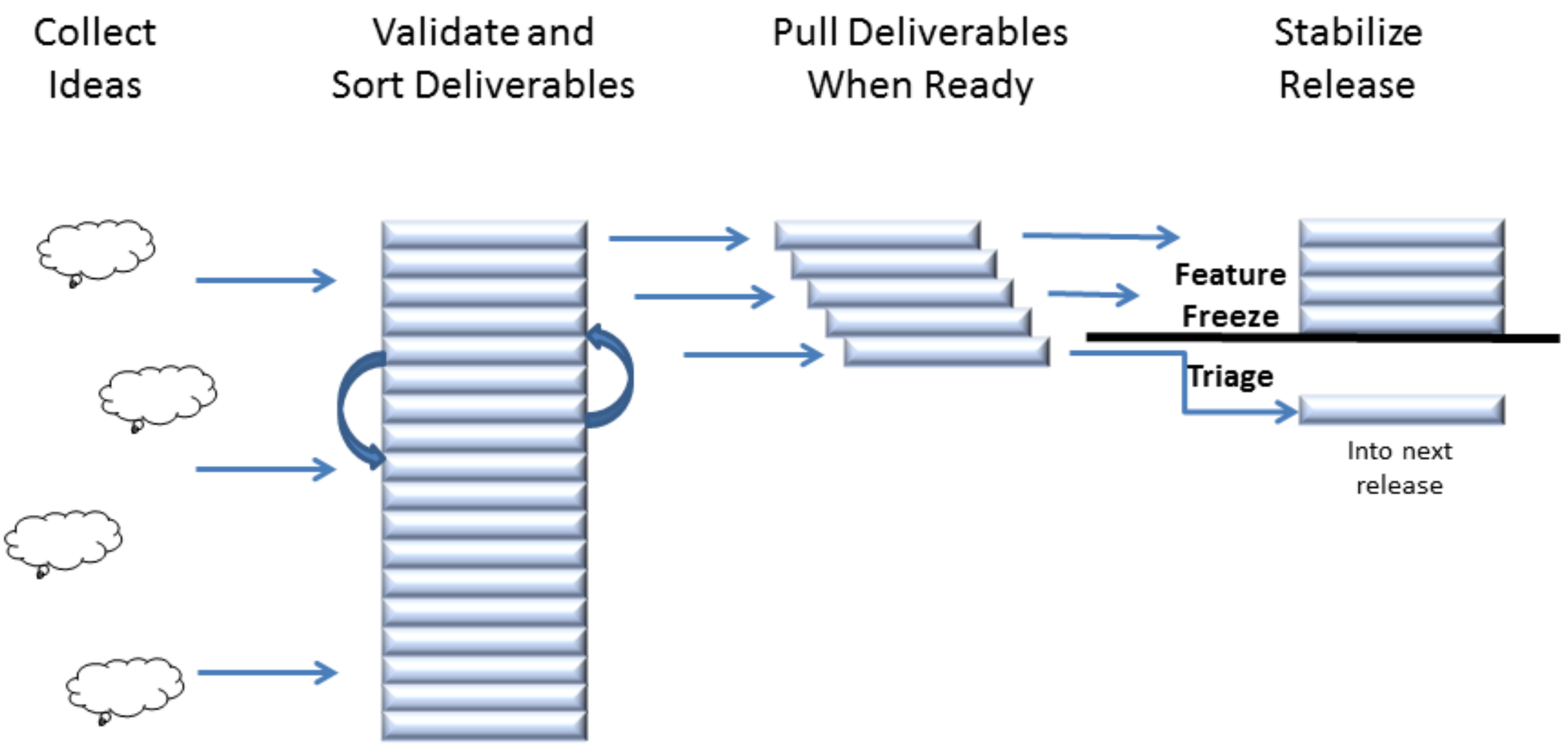
» Limit Work-in-Progress

» Feature freeze and stabilize

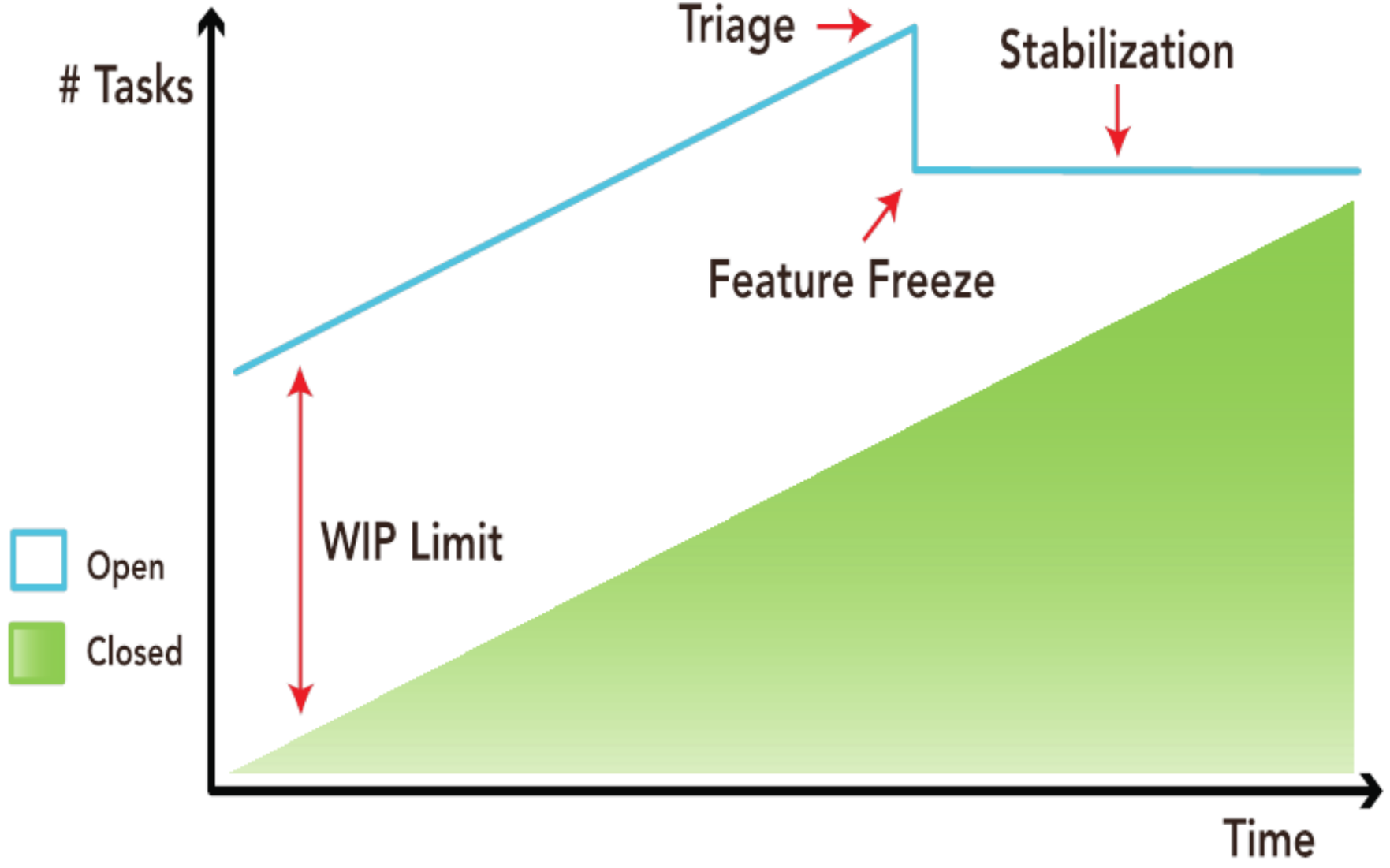
» Release and demo

» Iterate and reflect

Scrumban Iteration



Idealized ScrumBan Release



TEAM METRICS

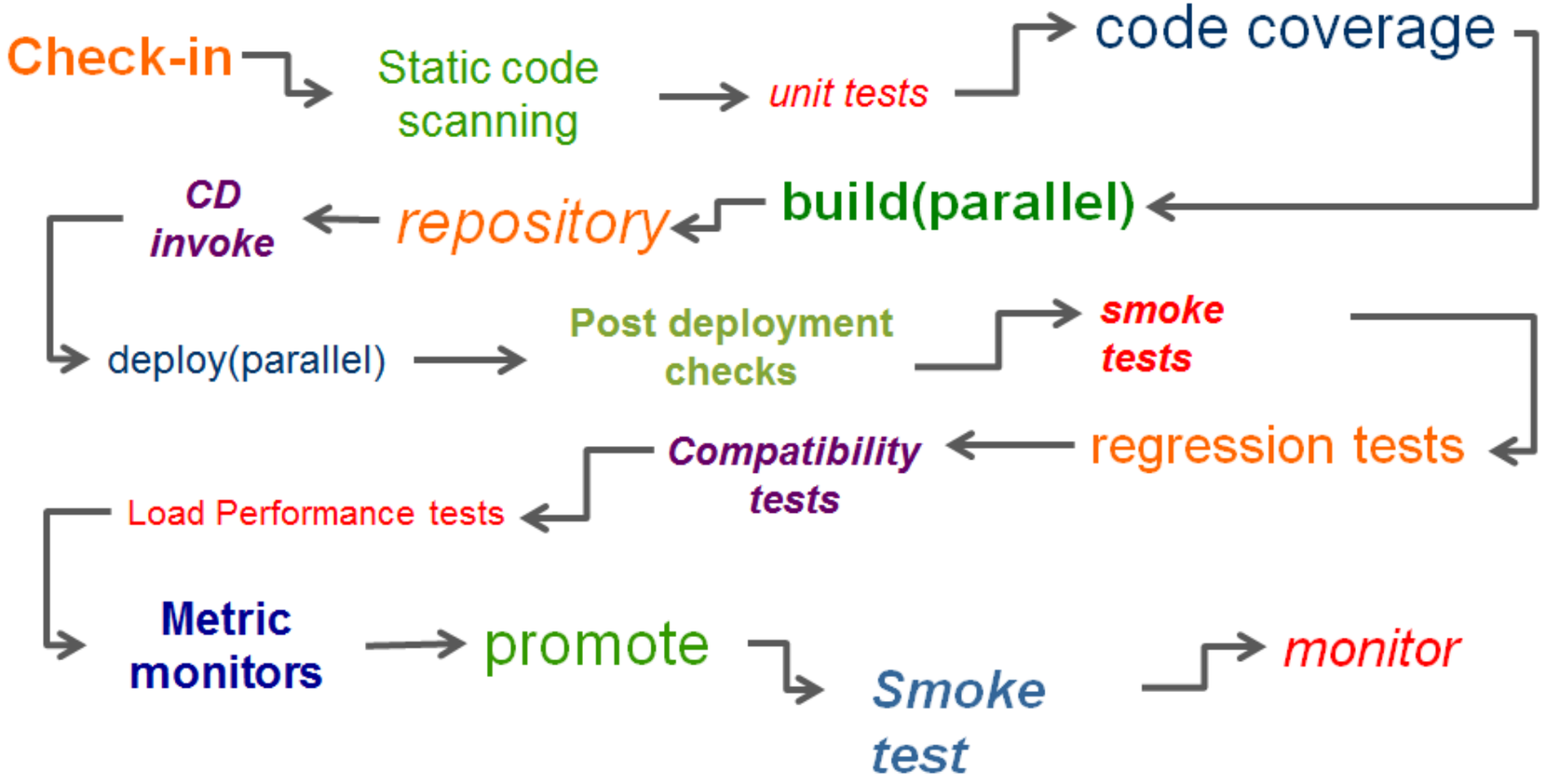
» Cycle time

» Velocity

» Story count

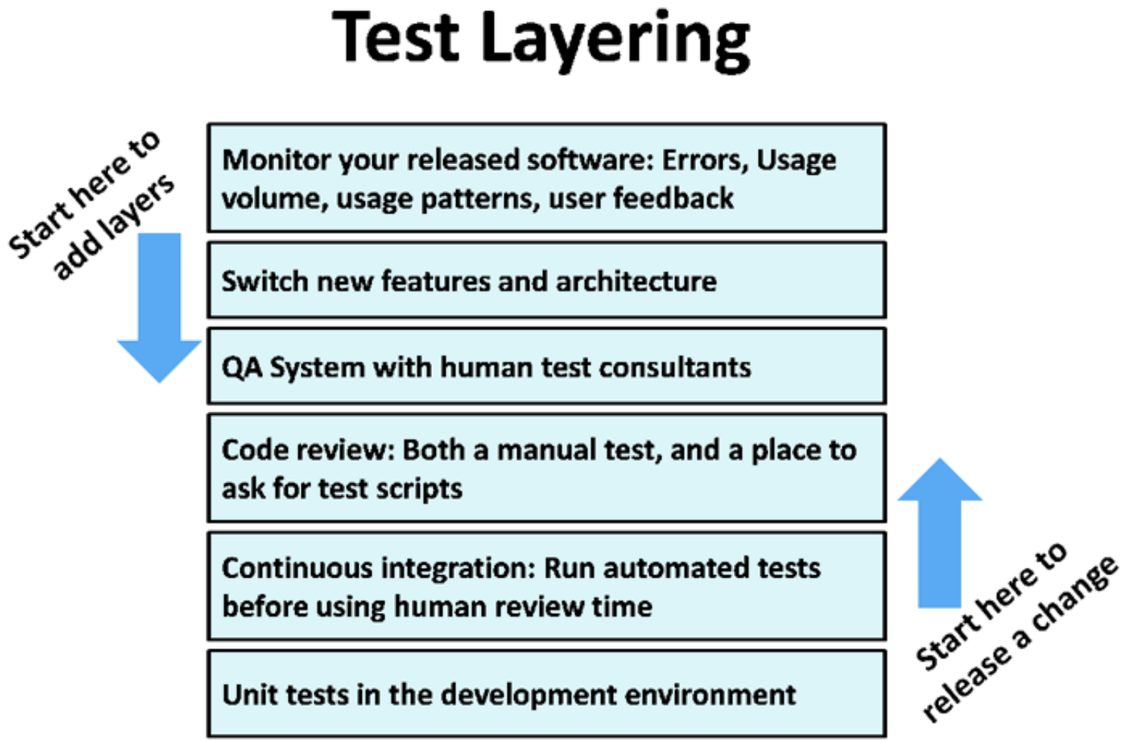
» Quality

ENGINEERING PRACTICES



AUTOMATED TESTING

- Mix automated and manual testing layers
- Any change is released only after passing all the test layers
- Automate as much as possible

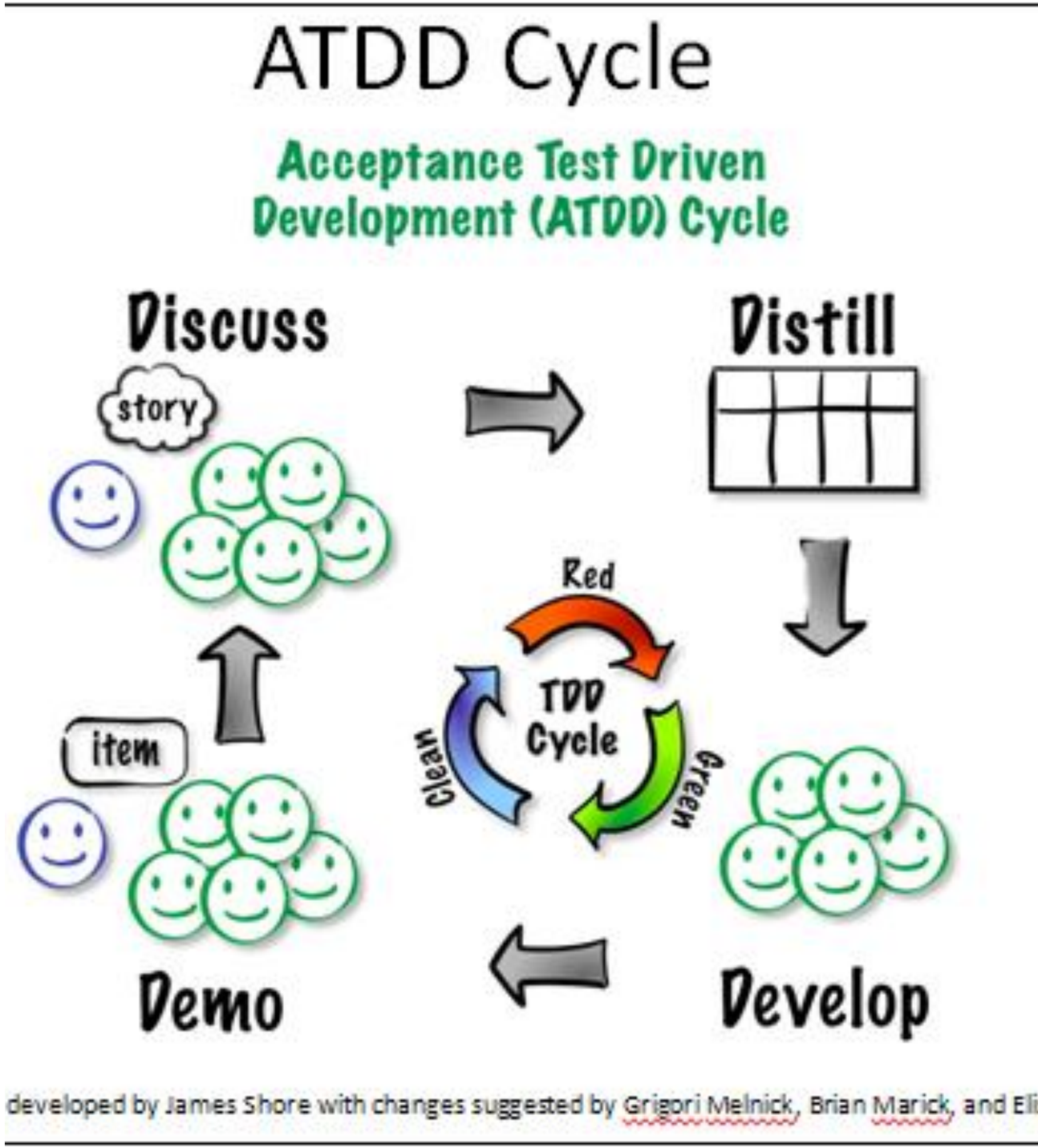


SPECIFICATION BY EXAMPLE

» Clear Acceptance Criteria

» Automated acceptance tests

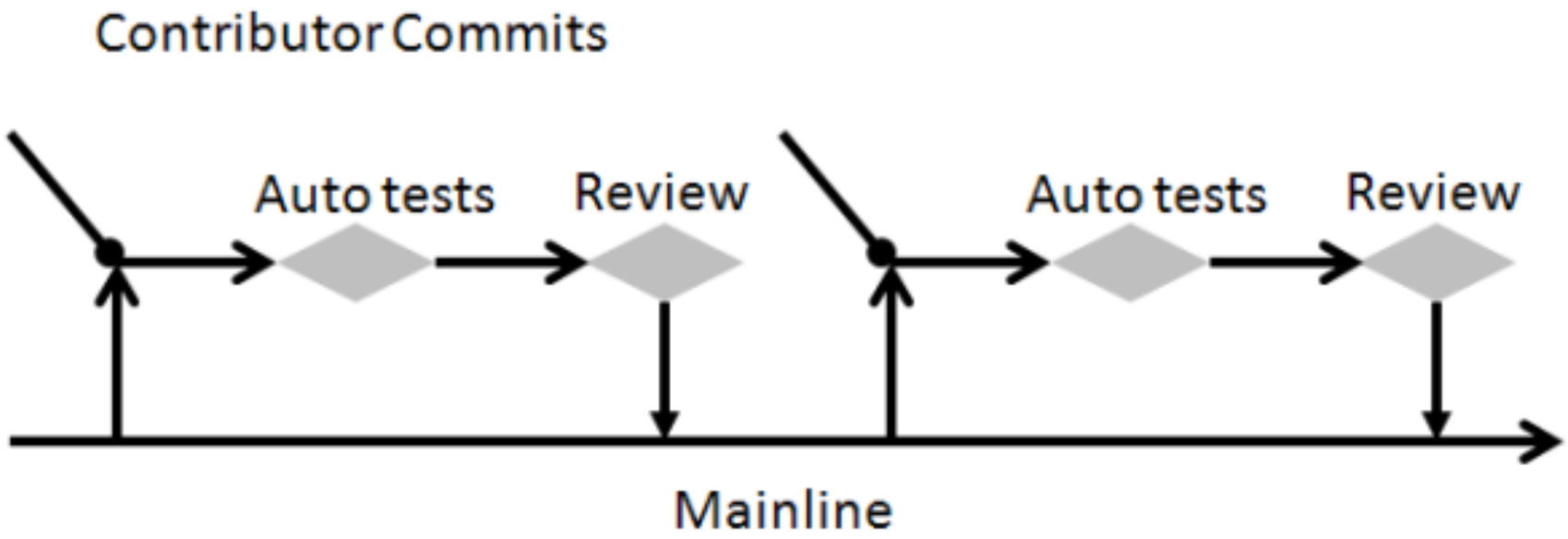
» Promote to regression tests



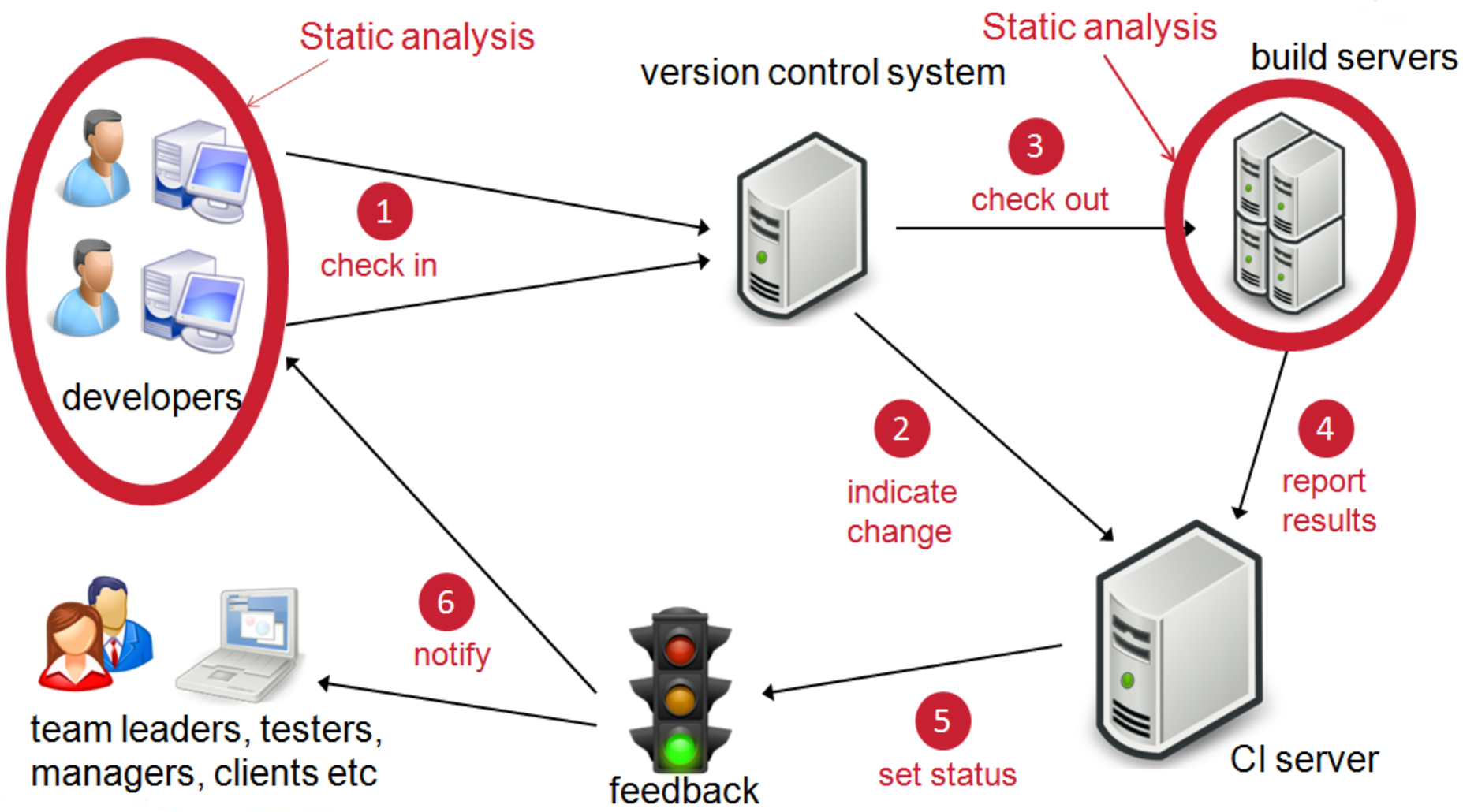
CODE REVIEW

» Static code analysis and unit tests before manual review

» Only quality code is merged to mainline



CONTINUOUS INTEGRATION



FEATURE SWITCHES

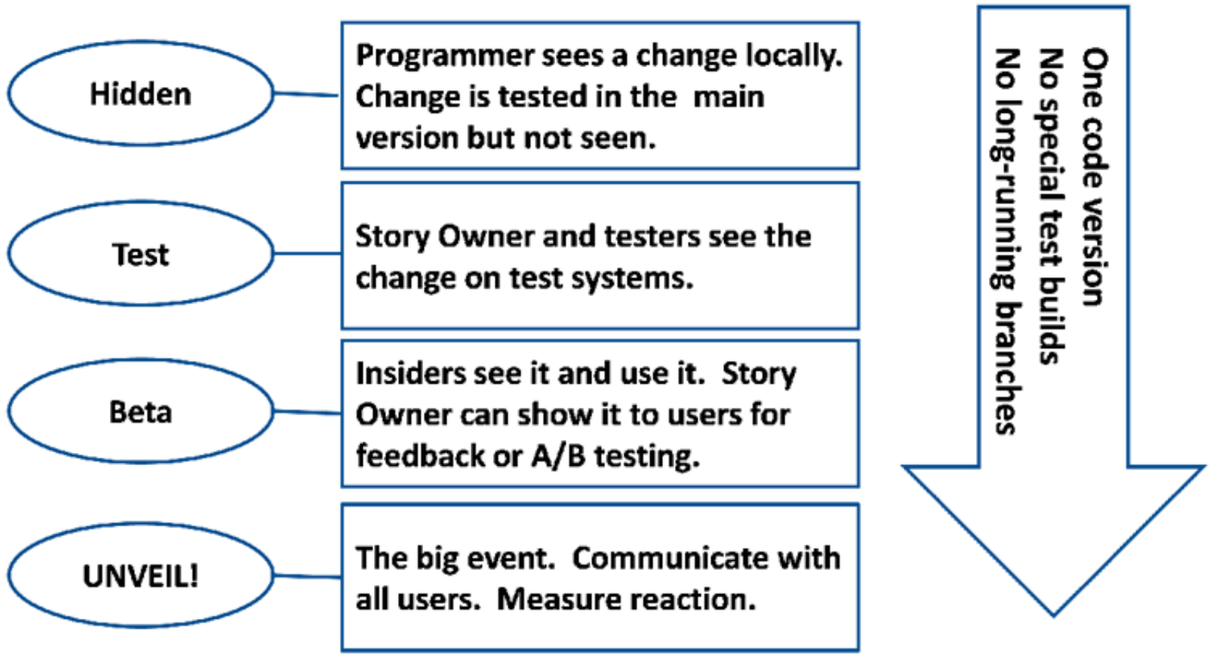
» Setup

» Install

» Overlap

» Unveil

» Remove



PUTTING IT ALL TOGETHER

» Build the right features

» Frequent feedback

» No process overhead

» No waste

» No technical debt

» Motivated teams

Q&A

EVGENY DEMCHENKO
CTO AT BESTOURS SOFTWARE