<epam>

Software development processes overview

KONSTANTIN ZLOBIN

Feb 2017

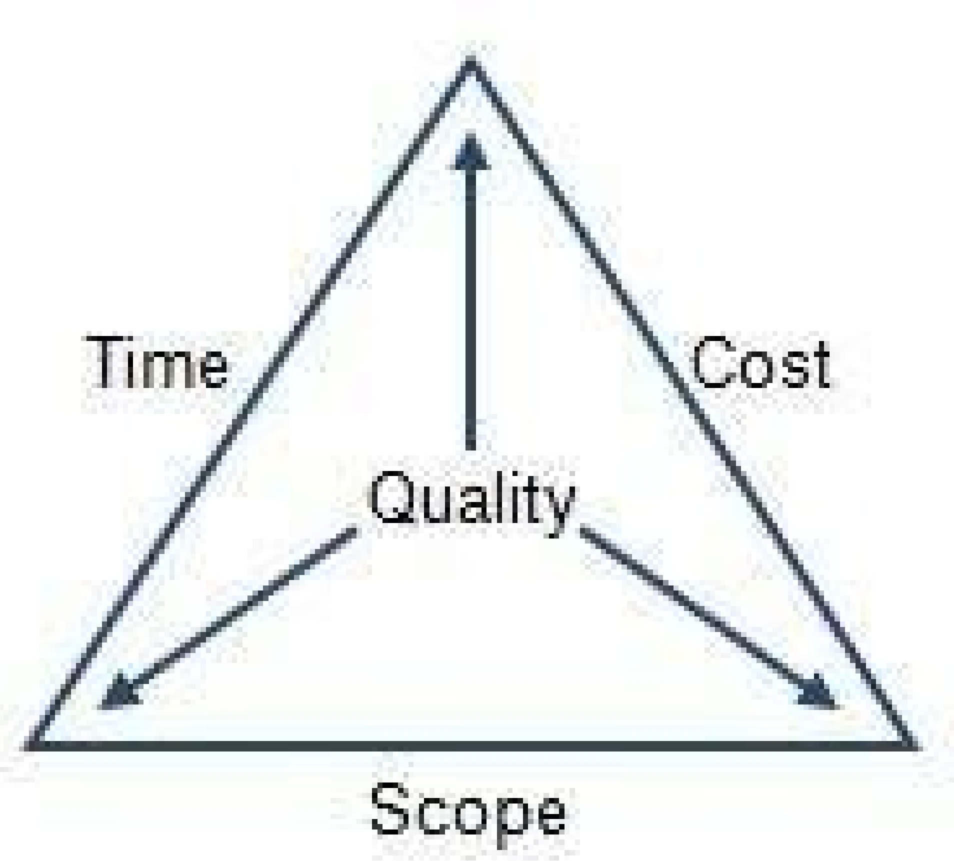
Types of software development processes

- Project oriented - generic
  - PMI - Project Management Institute
  - Project Manager, PMBok
  - Waterfall
- Service\flow oriented
  - Lean
  - Kanban
- Product oriented
  - Agile
  - Scrum
  - XP

A Project - what is it?

- A project is temporary in that it has a defined beginning and end in time, and therefore defined scope and resources.
- A project is unique in that it is not a routine operation, but a specific set of operations designed to accomplish a singular goal.
- A project team often includes people who don’t usually work together – sometimes from different organizations and across multiple geographies.
- The development of software for an improved business process, the construction of a building or bridge, the relief effort after a natural disaster, the expansion of sales into a new geographic market — all are projects.

Constraints



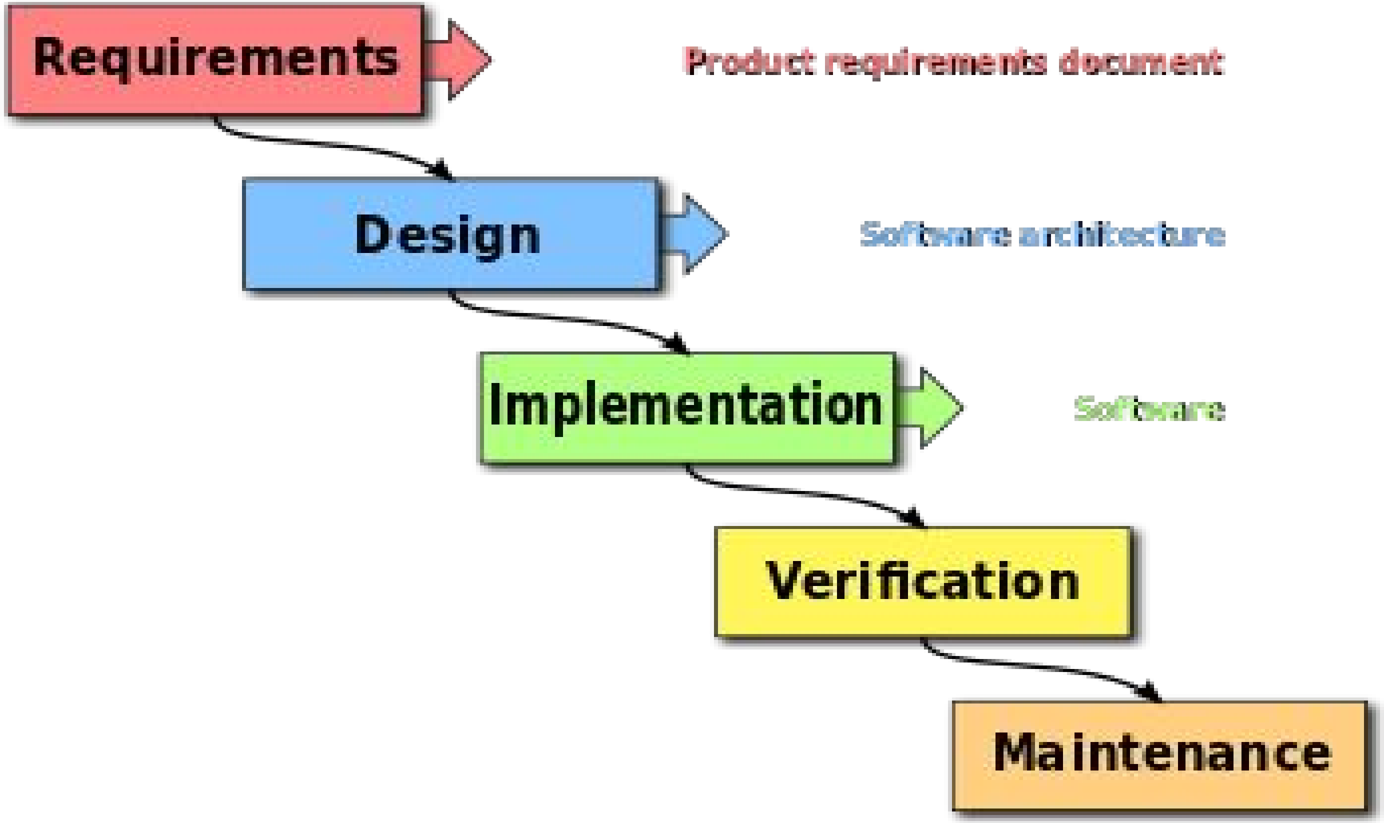
Project management processes (PMI)

1. Initiating
2. Planning
3. Executing
4. Monitoring and Controlling
5. Closing

Project management knowledge ten areas (PMI)

1. Integration
2. Scope
3. Time
4. Cost
5. Quality
6. Procurement
7. Human resources
8. Communications
9. Risk management
10. Stakeholder management

Waterfall. A very naive implementation of PMBoK



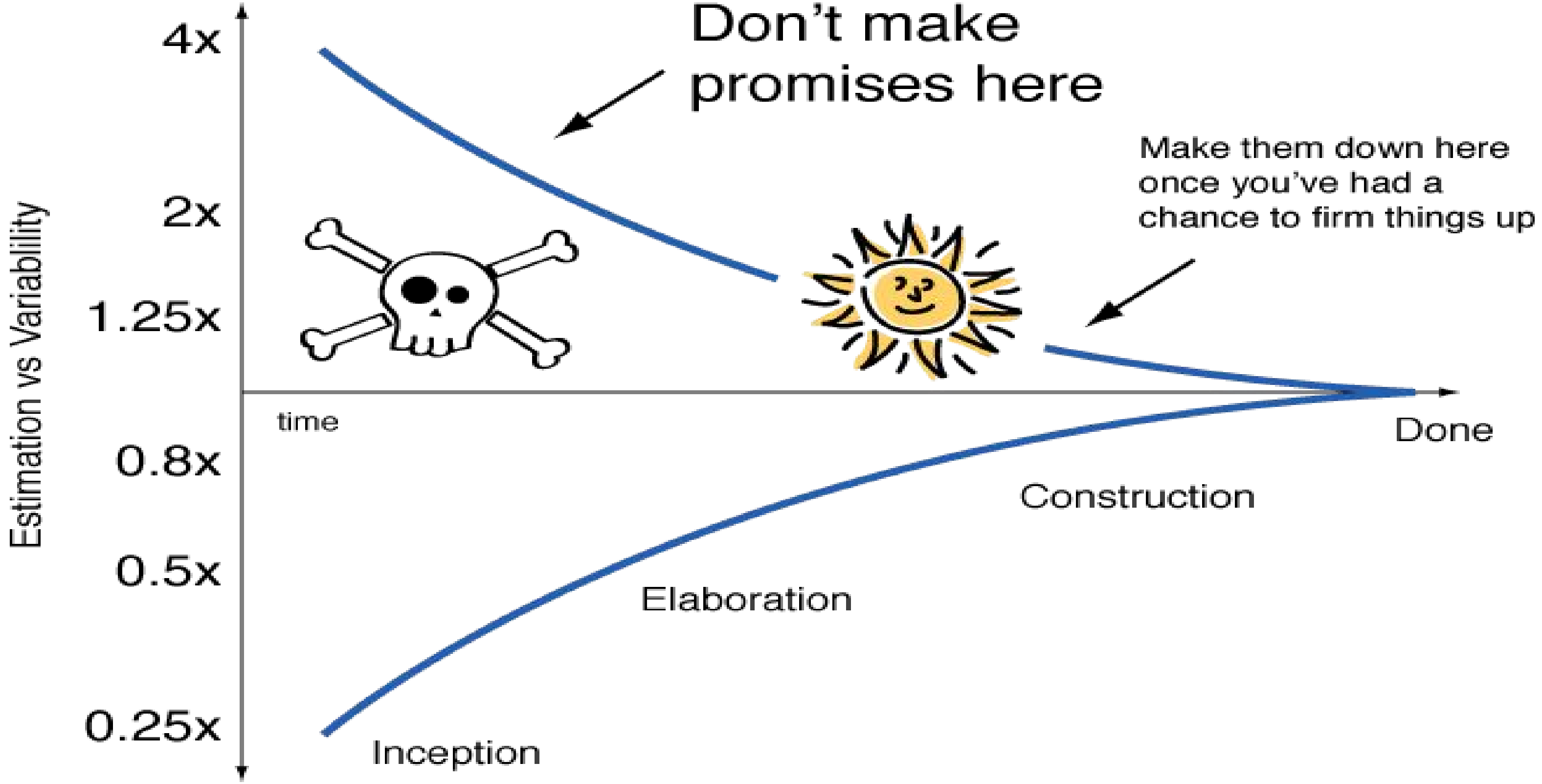
Waterfall. Pros

- PMI basic logic - first plan then do
- Provides a structured approach
    - easy to understand
    - easy to track
    - identifiable milestones
- Most risky phases are done first in less than 30% time of project
- Everything is documented
    - knowledge is not lost when team changes

Waterfall. Cons

- Clients don’t know what their requirements are before they see working software.
- Significant number of features of software product may be never used.
- Designers don’t aware of future difficulties when designing a new software product or feature.
- Poor quality of communication as the PM becomes the centre of communications and thus the bottleneck.
- People becomes irresponsible for the final product, because they don’t see the full picture.
- Low motivation in teams because of
    - too few feedback loops or none of them
    - bureaucracy

Cone of Uncertainty



Don’t make promises here

Make them down here once you’ve had a chance to firm things up

Waterfall. Where\When you might meet it

- Very large distributed team.
  - No chance people can talk to\understand each other - bureaucracy may be your best friend.
- Your customer is a big state organisation, not a private business.
  - They is lead by bureaucracy and you must speak their language
  - Users tolerate bad quality - they must use the product you build and cannot comply
- Requirements will not change significantly in the middle of the project.
  - The software product is not new to the market.
  - The project is about to customize well known product template.
- There is very small or no technology risk.
  - Developer organisation has competence in the technologies they are going to use
  - Market full of experts of the technologies you are going to use.

Project oriented process. Your role as a developer

- Know your role and play it well.
- There may be a lot of project specific policies\processes\rules you must know and follow. Ask your PM where to read them.
- Listen to PM of what to do.
- Do it.
- Report to your PM about results.

Lean manufacturing

- Manufacturing
  - Building something on a factory by human hands.
  - A lot of products of the same kind and features in a day.
  - Or service like a barbershop, software development activities.
- Lean
  - Comes from Toyota Production System
  - Just-in-time (JIT) manufacturing is a methodology aimed primarily at reducing flow times within production system as well as response times from suppliers and to customers
  - Smooth flowing delivery of value achieves all the other improvements as side-effects
- Elimination of Waste
  - Muda - waste in manufacturing system
  - Muri - waste through overburden
  - Mura - waste through unevenness in workload

Kanban

- Kanban is a method of organizing and managing professional services work. It uses Lean concepts such as limiting work in progress to improve results.
- A Kanban system is means of limiting work-in-progress and signaling when capacity is available to start new work. This is known as a “pull system.”
- For intangible goods — invisible work performed by knowledge workers — it is necessary to visualize a kanban system using a kanban board. The board can visualize the work, its workflow, the kanban system, plus various business and delivery risks. Kanban facilitates complex handling of the work and allows real control of the workload.

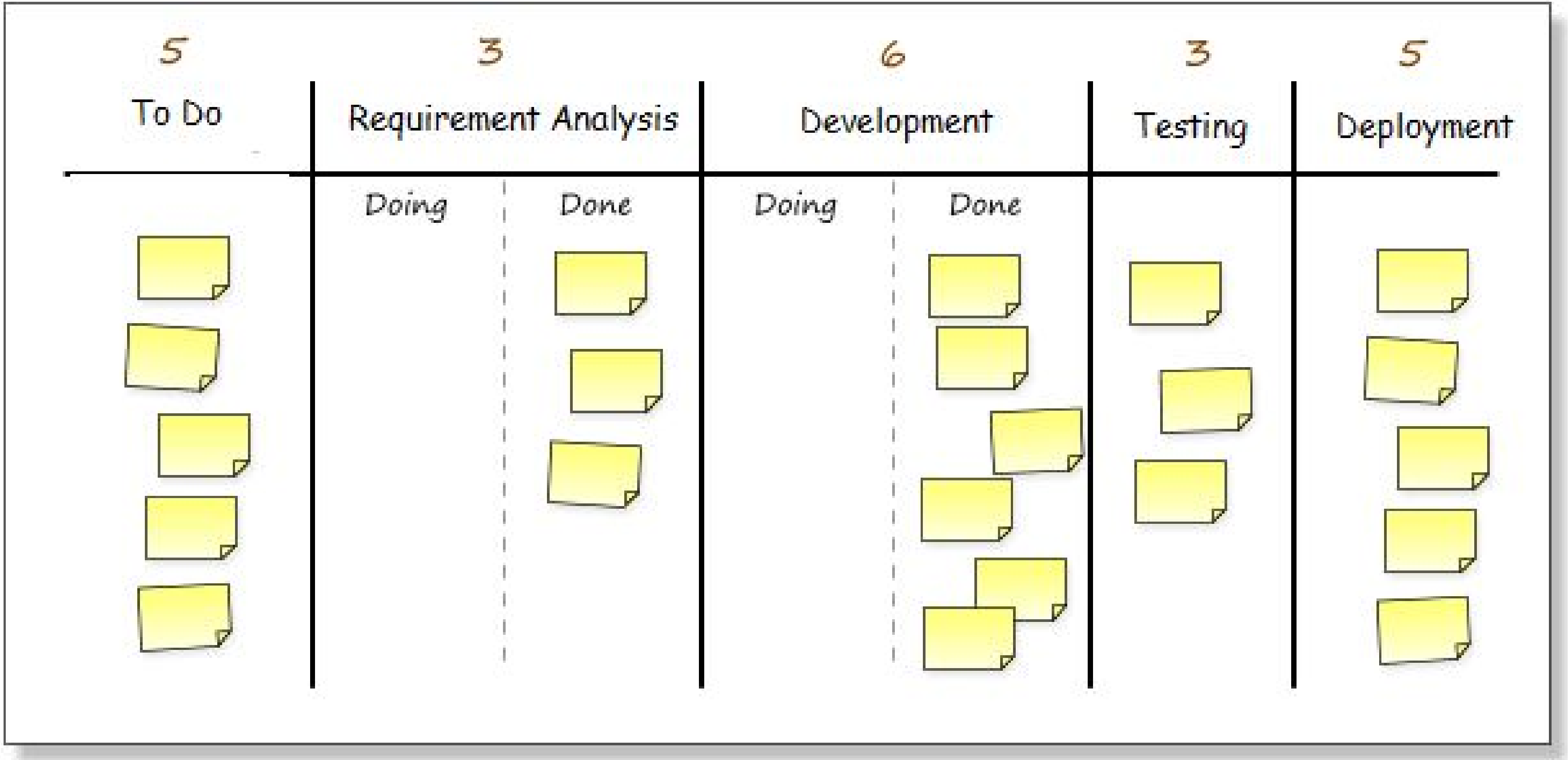
Kanban Method. Change management principles

1. Start with what you do now
2. Agree to pursue incremental, evolutionary change
3. Respect the current process, roles, responsibilities & titles
4. Encourage acts of leadership at all levels.

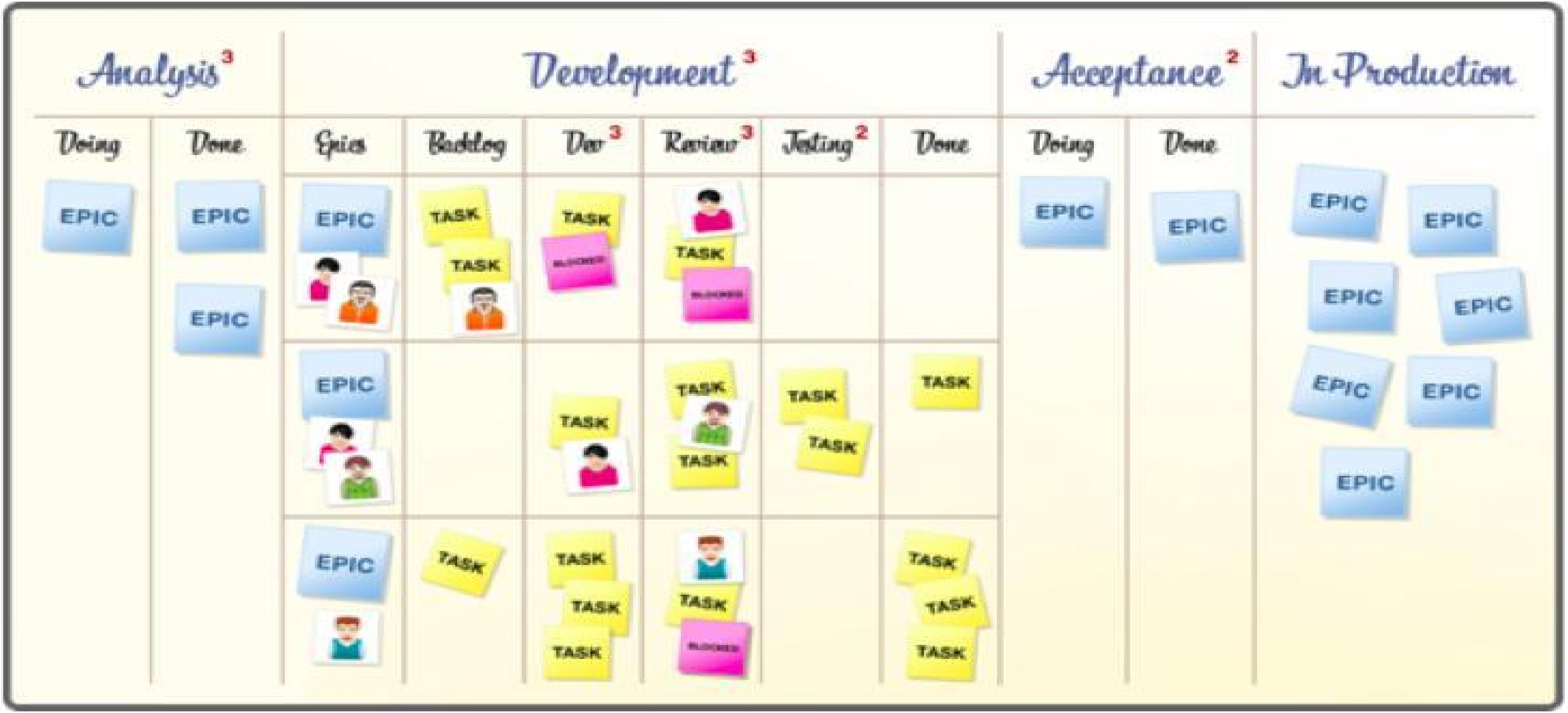
Kanban Method. Principles

1. Visualize the workflow
2. Limit WIP
3. Manage flow
4. Make Process Policies Explicit
5. Improve Collaboratively

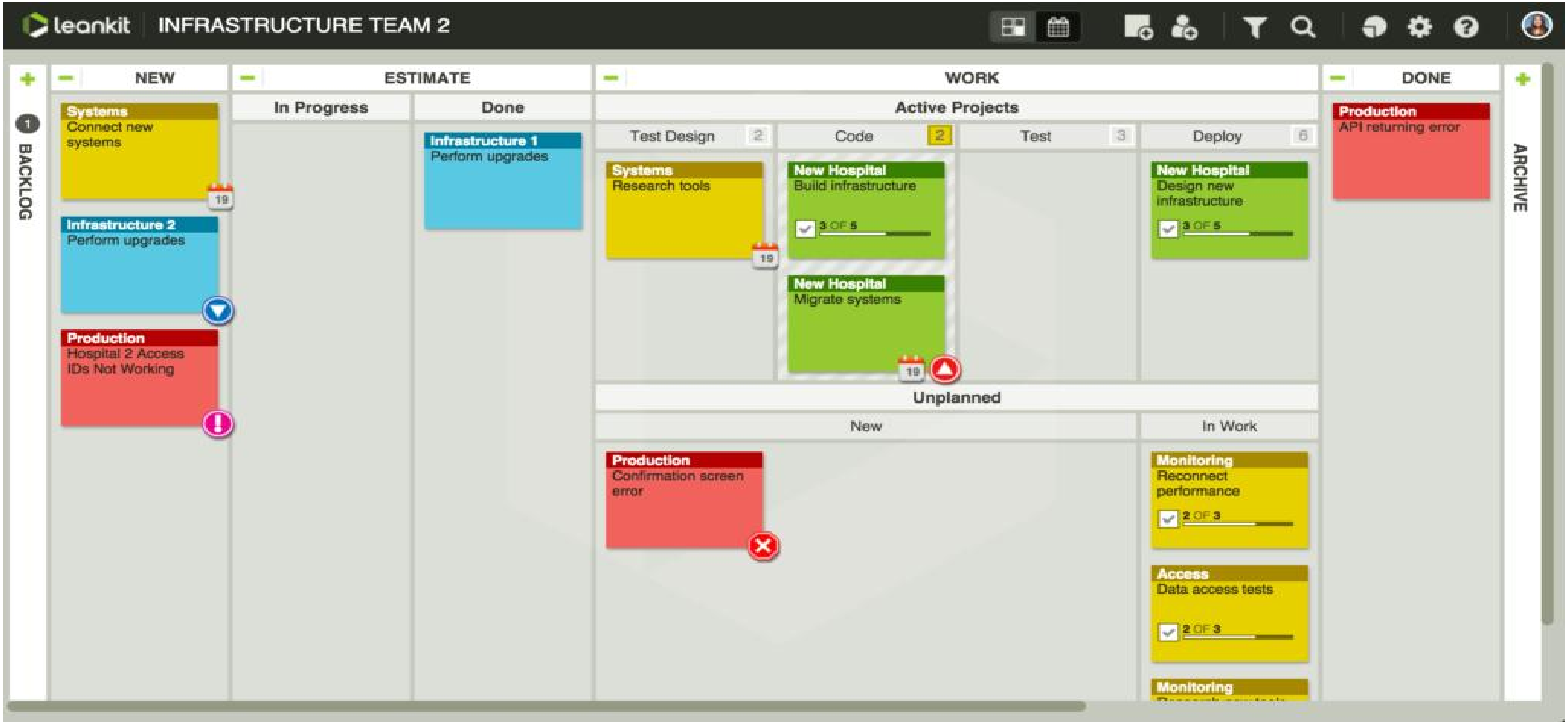
Kanban. Boards



Kanban. Boards



Kanban. Boards



leankit INFRASTRUCTURE TEAM 2

BACKLOG 1

NEW

Systems
Connect new systems
19

Infrastructure 2
Perform upgrades

Production
Hospital 2 Access IDs Not Working

ESTIMATE

In Progress

Done

Infrastructure 1
Perform upgrades

WORK

Active Projects

Test Design 2

Systems
Research tools
19

Code 2

New Hospital
Build infrastructure
3 OF 5

New Hospital
Migrate systems
19

Test 3

Deploy 6

New Hospital
Design new infrastructure
3 OF 5

Unplanned

New

Production
Confirmation screen error

In Work

Monitoring
Reconnect performance
2 OF 3

Access
Data access tests
2 OF 3

Monitoring

DONE

Production
API returning error

ARCHIVE

Kanban. Pros

- Waste elimination
- Answers on how to implement project management processes in many areas:
    - Change management
    - Risk management
    - Workflow management
    - Monitoring
    - Quality management
- Continuous delivery of value

Kanban. Cons

- Hard to predict when a project will be finished
  - Only statistical methods could be used
- Bad when the whole work cannot be break into independently deliverable pieces (a rocket to the moon)
  - Use Waterfall
- Bad when focus on team collaborative creativity is more important than even flow (early stages of product development).
  - Try Scrum

Kanban. Where\When you might meet it

- In a pure form
    - Software maintenance and support a.k.a. Bug-fixing
    - Advertisement campaigns support e.g. production of landing pages
    - Web-based product development on later stages when new features a rare and small
    - etc
- Partially used in Scrum-based processes
    - ScrumBan
- As a method to rebuild uneffective current process

Kanban. Your role as a developer

- Look at the most upper-right card on the board
  - Can you do something to move it to the next right column?
  - Do if you can (have qualifications needed).
  - Proceed with the next upper-right card on the board.
- When WIP limit is reached for “in development” column go and help somebody else with his or her work with another card from the board
- Don’t forget to update status of the cards on the board
- Follow explicit policies specified for actual column of the card
- Be a leader

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

That is, while there is value in the items on the right, we value the items on the left more

Product oriented. Agile. Scrum

- Scrum: A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.
- Scrum is:
  - Lightweight
  - Simple to understand
  - Difficult to master
- Scrum is founded on empirical process control theory, or empiricism.
- Significant aspects of the process must be visible to those responsible for the outcome.

The Sprint

- Is a time-box of one month or less during which a “Done”, useable, and potentially releasable product Increment is created
- Contain and consist of
  - Sprint Planning
  - Daily Scrums
  - development work
  - Sprint Review
  - Sprint Retrospective
- Each Sprint may be considered a project with no more than a one-month horizon.
  - Like projects, Sprints are used to accomplish something - Sprint Goal.
  - Sprints also limit risk to one calendar month of cost.

Scrum. Events

- Sprint Planning
  - What can be done this Sprint?
  - How will the chosen work get done?
  - Choose the Sprint Goal
- Daily Scrum
  - 15 min
  - 3 questions
- Sprint Review
  - Is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog
  - Scrum Team and stakeholders collaborate about what was done in the Sprint
- Sprint Retrospective
  - Is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint

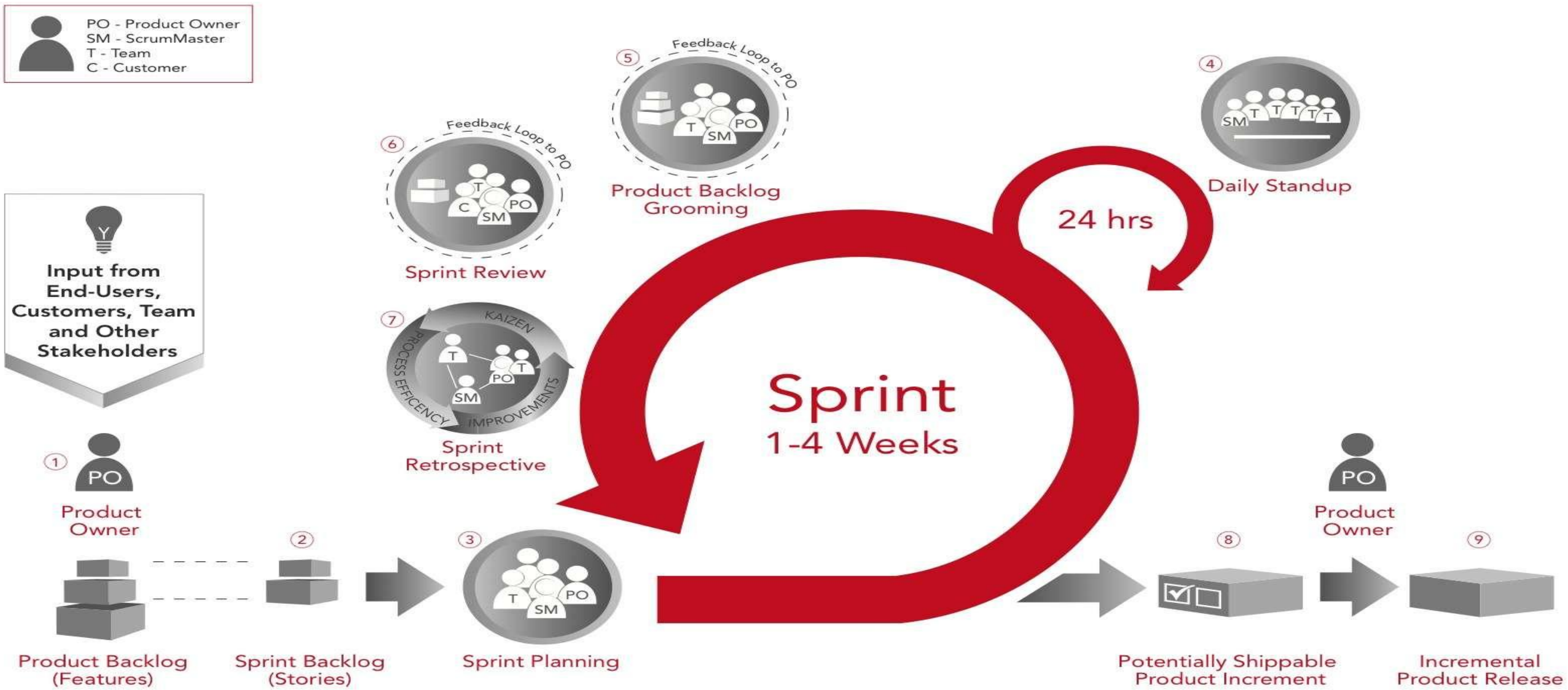
The Scrum Team

- Product Owner
  - maximizing the value of the product
  - manages the Product Backlog
  - one person, not a committee
  - organization must respect his or her decisions
  - Development Team isn’t allowed to act on what anyone else says
- Development Team
  - self-organizing and cross-functional
  - is small enough to remain nimble and large enough to complete significant work within a Sprint
- Scrum Master
  - Scrum Master is responsible for ensuring Scrum is understood and enacted
  - Scrum Master is a servant-leader for the Scrum Team

Scrum. Artifacts

- Product Backlog
  - is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product
  - is never complete
  - higher ordered items are usually clearer and more detailed than lower ordered ones
  - Development Team is responsible for all estimates
- Sprint Backlog
  - Set of Product Backlog items
  - Plan on how to meet the Sprint Goal and deliver the Increment
- Increment
  - is the sum of all the Product Backlog items completed during a Sprint
  - potentially shippable - meet the Scrum Team’s definition of “Done”

How Scrum Works



PO - Product Owner
SM - ScrumMaster
T - Team
C - Customer

Input from End-Users, Customers, Team and Other Stakeholders

1 Product Owner

Product Backlog (Features)

2 Sprint Backlog (Stories)

3 Sprint Planning

4 Daily Standup

24 hrs

5 Product Backlog Grooming

6 Sprint Review

7 Sprint Retrospective

Sprint
1-4 Weeks

8 Potentially Shippable Product Increment

Product Owner

9 Incremental Product Release

scruminc.

Scrum. Pros

- Transparency, inspection, and adaptation come to life and build trust for everyone
- Ensures effective use of time and money
- Projects are divided into easily manageable sprints
- Team gets clear visibility through scrum meetings
- It adopts feedback from customers and stakeholders
- It is easy to change based on feedback

Scrum. Cons

- Project failure are high if individuals aren't very committed or cooperative
  - Hire responsible people
- Scrum in large teams is challenging
  - 6+\-3 - the right size, also try SAFe\Nexus\Scrum of Scrums
- Can be successful only with experienced team members
  - Team must be cross-functional - enough qualifications to produce an increment
- If any team member leaves the project in the middle, it has a huge negative impact on the project
  - Know team formation logic: Storming, Norming, Performing
- Scrum is expensive
  - Don’t practice it when you don’t have clear verified product idea and vision

Scrum. Where\When you might meet it

- Active product development
  - The right place to practice the Scrum
- Formal elements of it “Scrum, but…” in any kind of project
  - Usually Scrum is badly implemented here and is not working

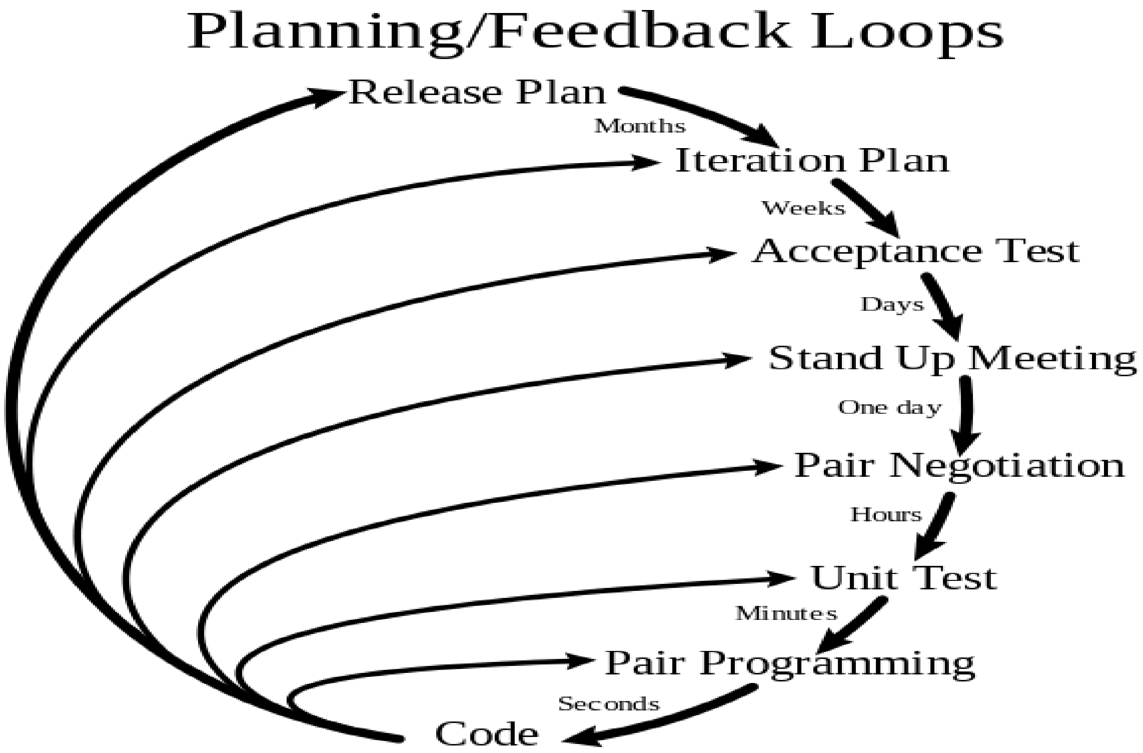
Scrum. Your role as a developer

- Be responsible
- Be a leader
- Work as as a team member
- Communicate and collaborate
- Pursue the Sprint Goal
- Don’t be late to meetings

Agile. eXtreme Programming (XP). Principles

- Planning game, User Stories
- Small releases
- System Metaphor
- Simple design
- Test Driven Development
- Refactoring
- Pair programming
- Continuous integration
- Collective code ownership
- On-site customer
- Coding standards
- 40-hour work week

eXtreme Programming (XP). Feedback Loops



XP. Where\When you might meet it

- In a pure form - European product companies
- Separate practices like TDD, Pair Programming - any project

Processes Symphony

- Scrum can be implemented as a sequence of Waterfalls. One waterfall per Sprint
- Scrum actively uses Kanban boards
- Limiting WIP is useful in any process
- Any modern software development process usually uses XP practices:
    - Continuous Integration - is a must!
    - Test Driven Development - everyone says it is cool but rarely follow
    - Concept of User Story in Scrum
    - Other
- PMBok is still valid in any process - it’s point of view is useful
- Traditional PMI Project Managers could benefit from self-organization in teams and a concept of flow management

Links and Literature

- PMI - https://www.pmi.org
- Lean Kanban University - http://edu.leankanban.com
- Agile Manifesto
  - 1.0 - http://agilemanifesto.org
  - 2.1 - https://agilescout.com/agile-manifesto-2-1-moreagile-manifesto
- Scrum Guide (bible :)) - http://www.scrumguides.org/scrum-guide.html
- https://www.scrum.org/ - certifies PSM, PSPO
- https://www.scrumalliance.org - certifies CSM, CSPO
- Extreme Programming Explained: Embrace Change, 2nd Edition by Kent Beck

Q&A