Backlog Management

Prepared by Gemini Solutions for FMI Bucharest

26.02.2018

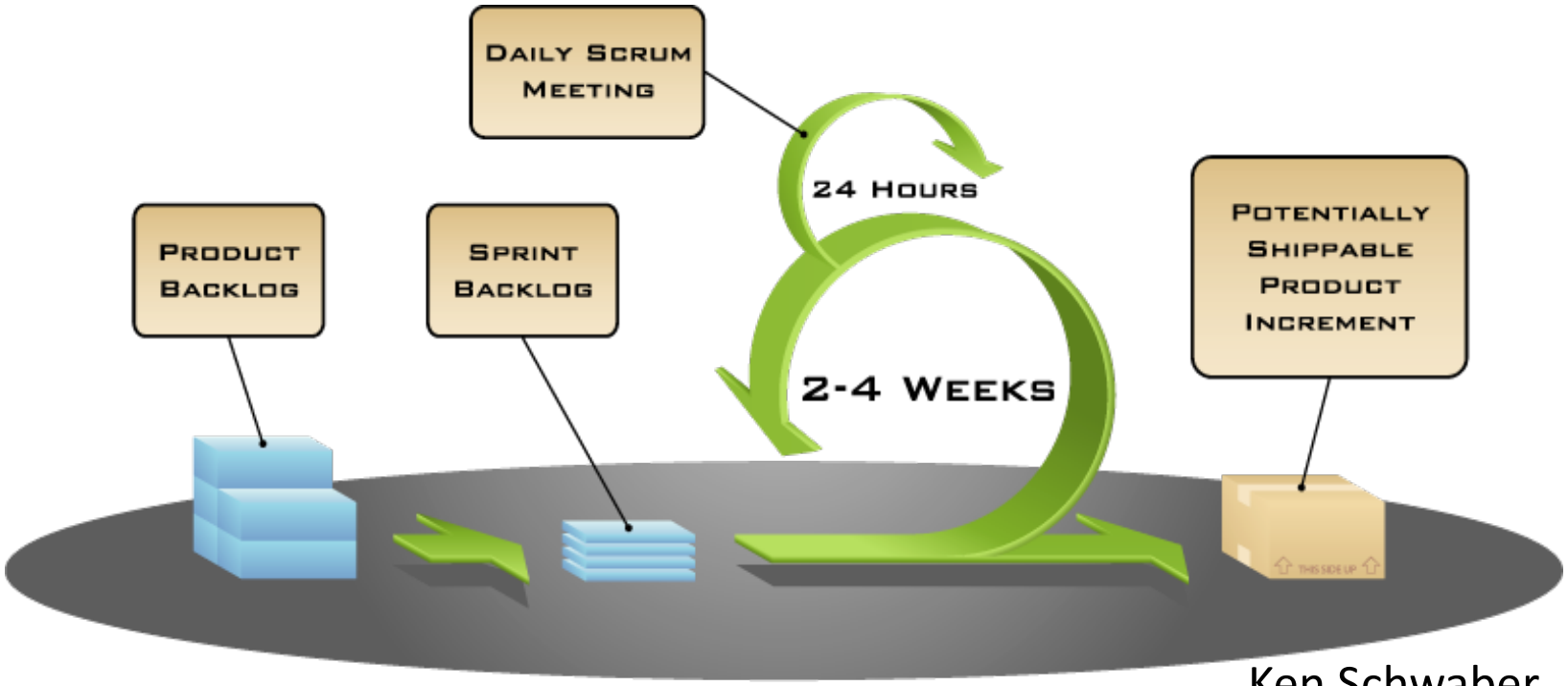
Some ways to express requirements

- Product Requirements Document (PRD)
  - Lots of chapters, sections, paragraphs – includes signoff section

- Use cases
  - Summary
  - Rationale
  - Users
  - Preconditions
  - Basic (happy) flow
  - Alternate paths 1..N

- User stories

Scrum framework



Ken Schwaber

Typical backlog contains…

- Epics
- User stories
- Non-functional requirements
- Chores
- Defects

So a user story…

- Basic unit of work in an agile project
- Describes a desired piece of business functionality
- Small enough to be implemented in an iteration
- A good user story is the simplest statement about the system that:
  - The customer cares about
  - Test cases can be written to verify
  - Can be reasonably estimated
  - Can be reasonably prioritized

Examples

Generic users stories

As a user, I want to reserve a hotel room

As a user, I want to cancel a reservation

Specific users stories

As a vacation planner, I want to see photos of the room

As a frequent flyer, I want to rebook a past trip, so that I save time booking

- Identify clearly the different users and their expectations …
- … especially for primary users (work on the right priorities)

Where are the details ?

- An example from a job posting and search website

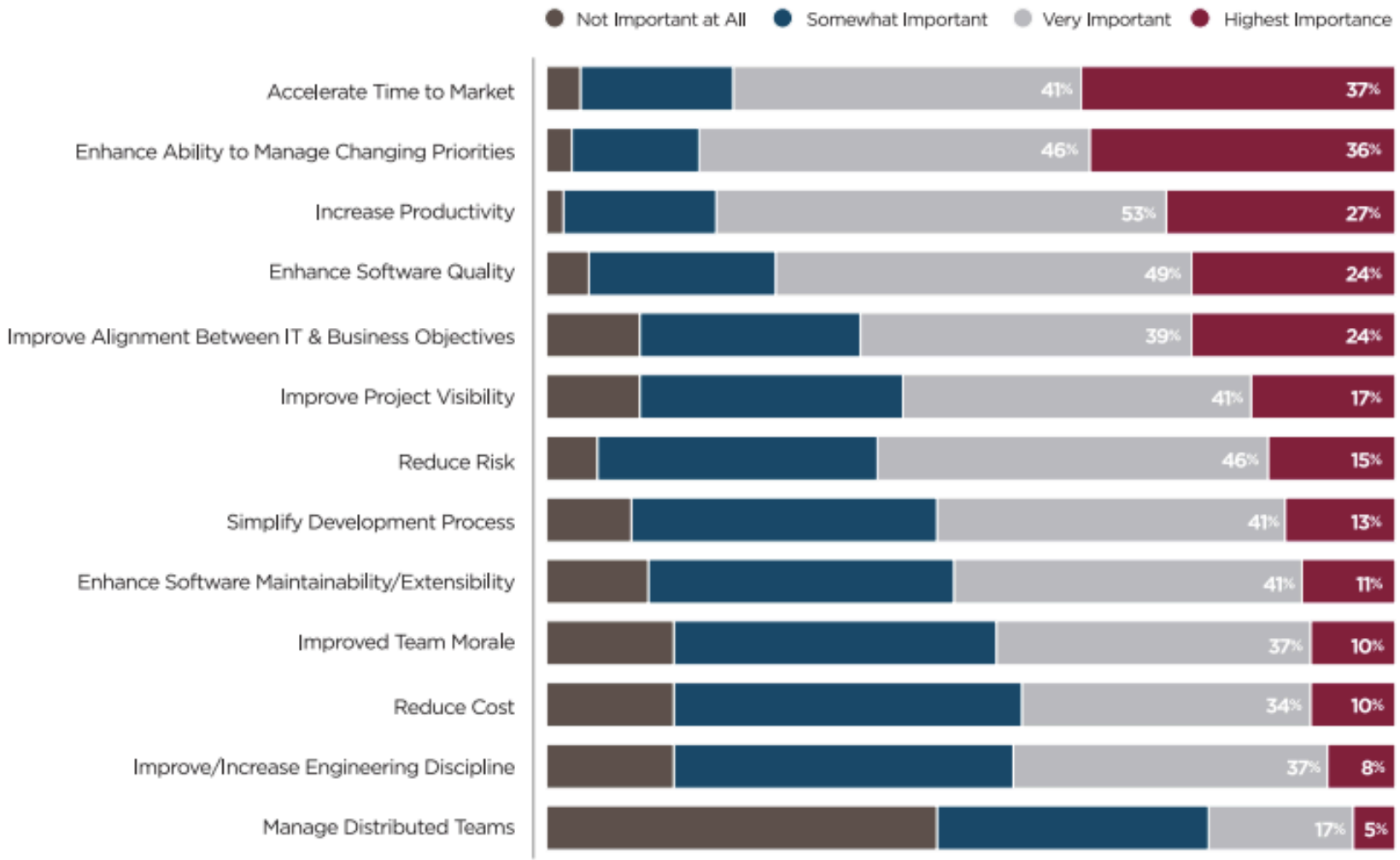
  A user can post her resume to the website.

  - What values can users search on? Country? City? Job title?
  - Does the user have to be a member of the site?
  - Can search parameters be saved?

- Many of these details can be expressed as additional stories

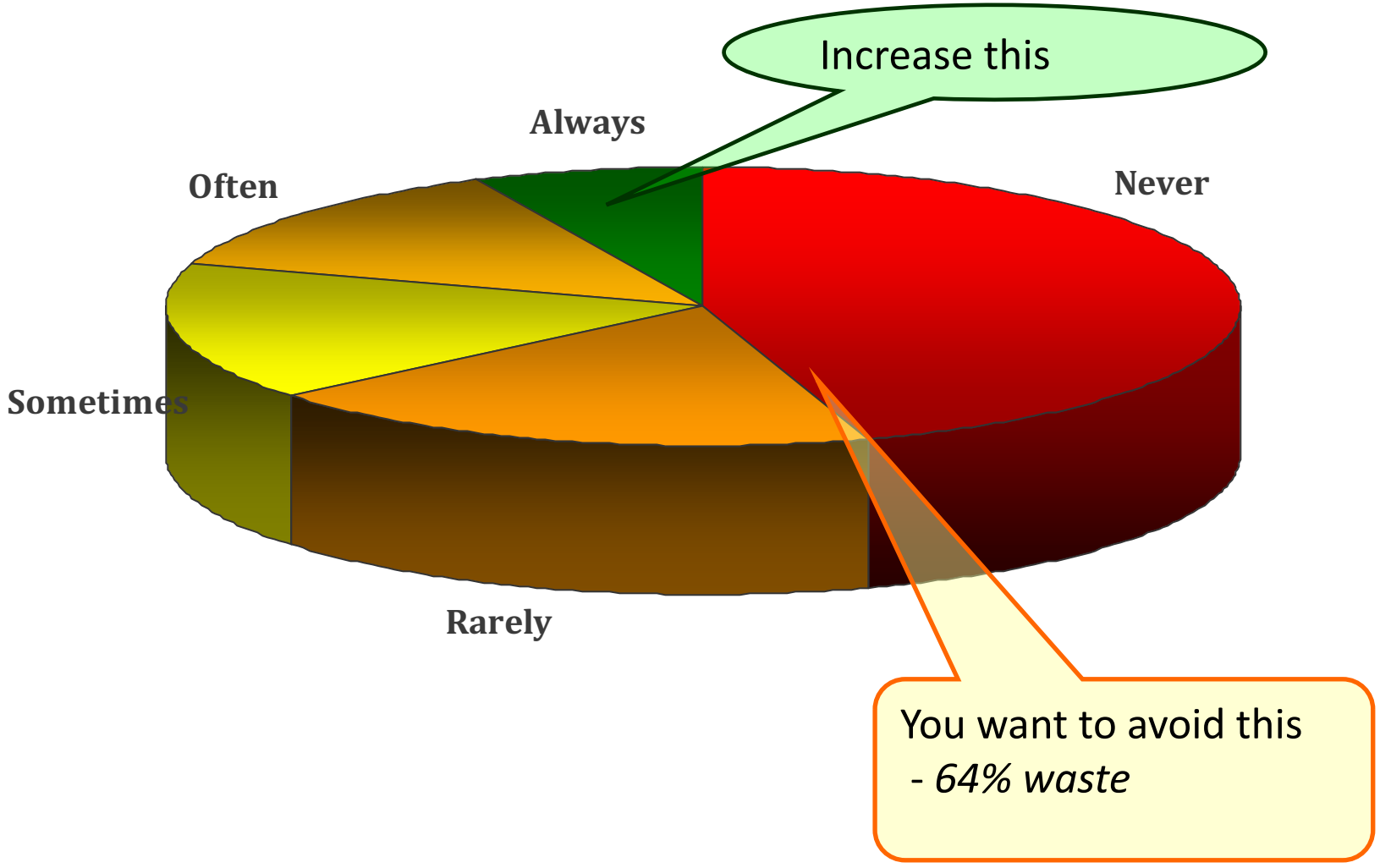
- A too large user story is called an *epic* and can be split into stories of smaller size

Industry reasons to adopt agile



VERSIONONE
Simplifying Software Delivery

Big Requirements Up Front (BRUF) – feature use



Source: Chaos Report v3, Standish Group, Copyright 2005-2006 Scott W. Ambler

Two important thoughts in Backlog Management

- Capitalize with the 80/20 Rule
  - 20% of the features are used 80% of the time
  - Finish that 20% first and put it into use, providing quick returns
  - Re-evaluate remaining requirements

- Accommodate Change!
  - Allows Business to drive the projects
  - Business can add new requirements and reprioritize existing ones
  - Short list of organizational limitations to changes

Personas

- Maria e studenta la Mate-Info la Bucuresti. Locuieste in Otopeni si ia in fiecare zi autobuzul 449 pana la Piata Presei, unde conecteaza cu una din liniile 305, 331 sau 131 pana la Charles de Gaulles, de unde ia linia 2 de metrou pana la Universitate. Nu-i place sa stea la inevitabila coada pentru abonamente studentesti si e stresata ca in statia ei din Otopeni nu poate sa cumpere bilete. Nu-i place sa stea iarna in frig in statie, dandu-se ca autobuzul respecta arareori programul.

Personas

- Pavel este avocat, locuieste in Bucuresti in Drumul Taberei si merge cu masina frecvent la client. Este dependent de masina. I se pare ca traficul in Bucuresti este ingrozitor, iar parcarea este foarte stresanta. S-a saturat de amenzi de parcare si i-a fost ridicata masina de trei ori pana acum.

INVEST in good stories

INVEST in good stories

- Independent
  - Avoid introducing dependencies (leads to difficulty prioritizing and planning)
- Negotiable
  - Stories are not written contracts
  - Do not need to include all details
- Valuable
  - Valuable to users and customers
- Estimable
  - Because stories are used in planning
- Small
  - Large stories are hard to estimate
- Testable
  - Tests demonstrate that a story meets the customer’s expectations
  - Automate if possible

Definition of estimate

- Estimate

- To form an **opinion** about
- A **tentative** evaluation or **rough** calculation
- A statement of the **approximate** cost of work to be done

- Accurate

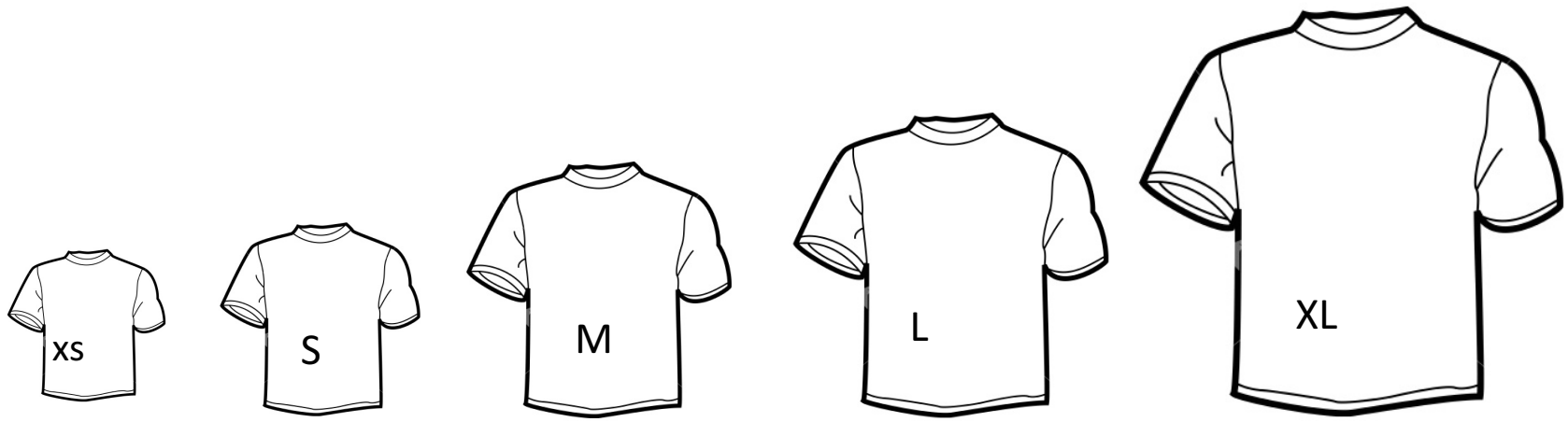
- Conformity to **fact**
- **Precision**
- **Exactness**
- Representation of the **truth**

Estimate != Accurate

What to consider when estimating

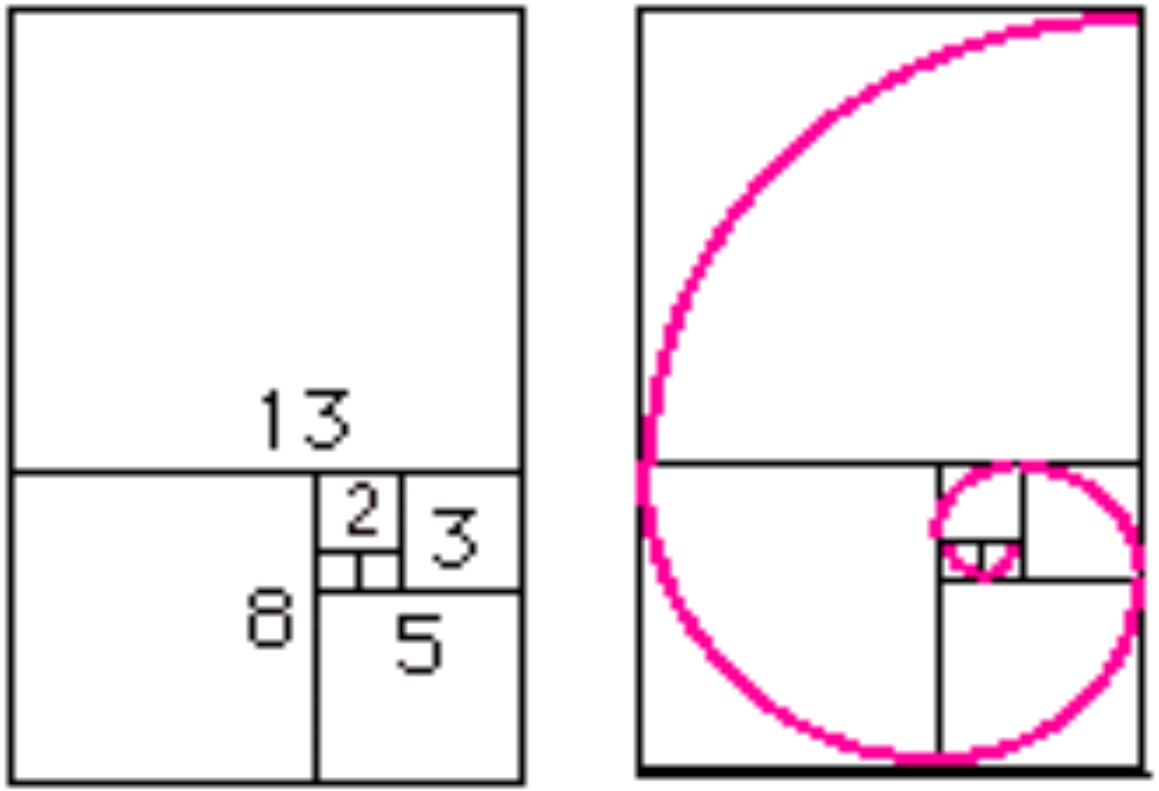
- Tests
  - Unit test for each line, acceptance test for each story
- Refactoring
  - The first time duplication or inefficiency is a concern
- Infrastructure
  - Considered for the first implementation of each layer
- Complexity
  - Some tasks are tough, and need more time (unknowns?)
- Tedium
  - Some tasks just take long because they do
- Everything…
  - Roll up all considerations into a single unit of measure

T-shirt sizes



XS – Trivial, almost no effort and is very well known
S – Low complexity and effort, known and well understood
M – More complexity and/or more effort than a Small
L – High complexity and/or effort. These stories may be broken down closer to the time they are played
XL - Too Big. An epic story with lots of unknown. These stories need more attention

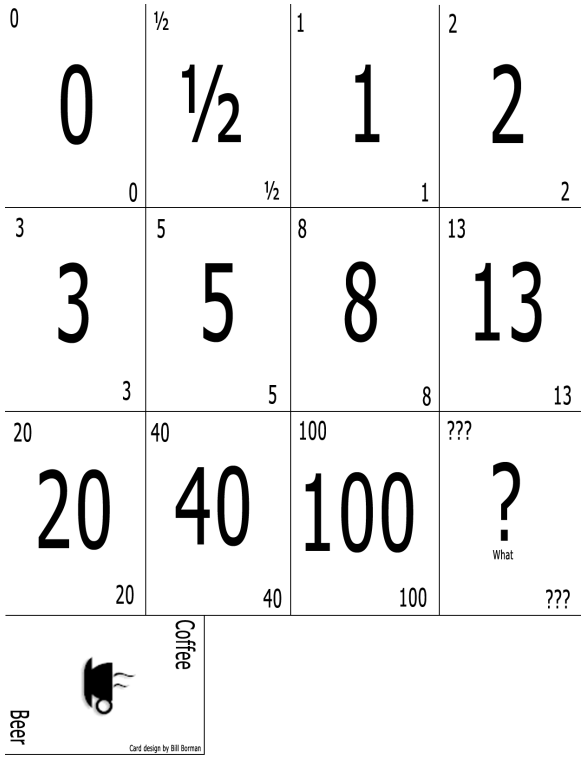
Fibonacci



Who estimates ?

One way to estimate: Planning Poker

Planning Poker = Fibonacci + Playing Cards



Planning poker rules

- Product Owner / Manager / RA explains story
- Team discusses work involved
- Everyone estimates individually
- Everyone reveals estimates simultaneously
- Lowest and highest estimates are justified
- Repeat until estimates converge

Overconfidence in estimation

- 90-90 rule
  - The first 90 percent of the code accounts for the first 90 percent of the development time. The remaining 10 percent of the code accounts for the other 90 percent of the development time.

– Tom Cargill, Bell Labs

- Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law.

- Douglas Hofstadter: Gödel, Escher, Bach: An eternal golden braid

Prioritization levels

- Releases
  - Milestones
    - Iterations
      - Order inside an iteration

Prioritizing the backlog: MoSCoW

- Must – a feature without which you don’t have a product
- Should – a critical requirement that should be released
- Could – desirable but not necessary
- Won’t – may be considered for the future

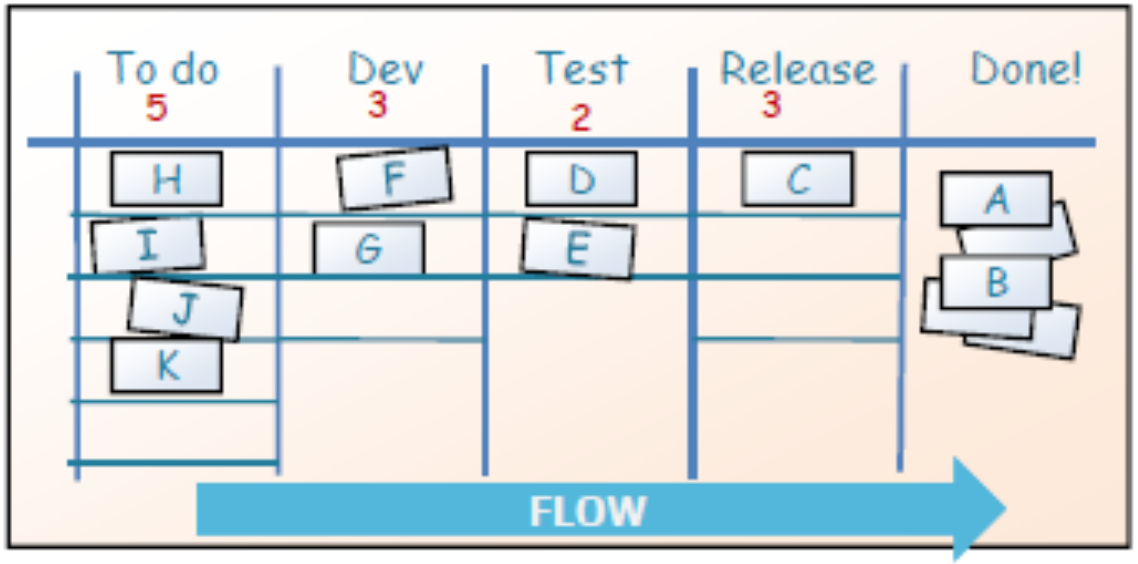
Velocity and how estimate it

- Velocity = sum of estimates of done user stories in an iteration

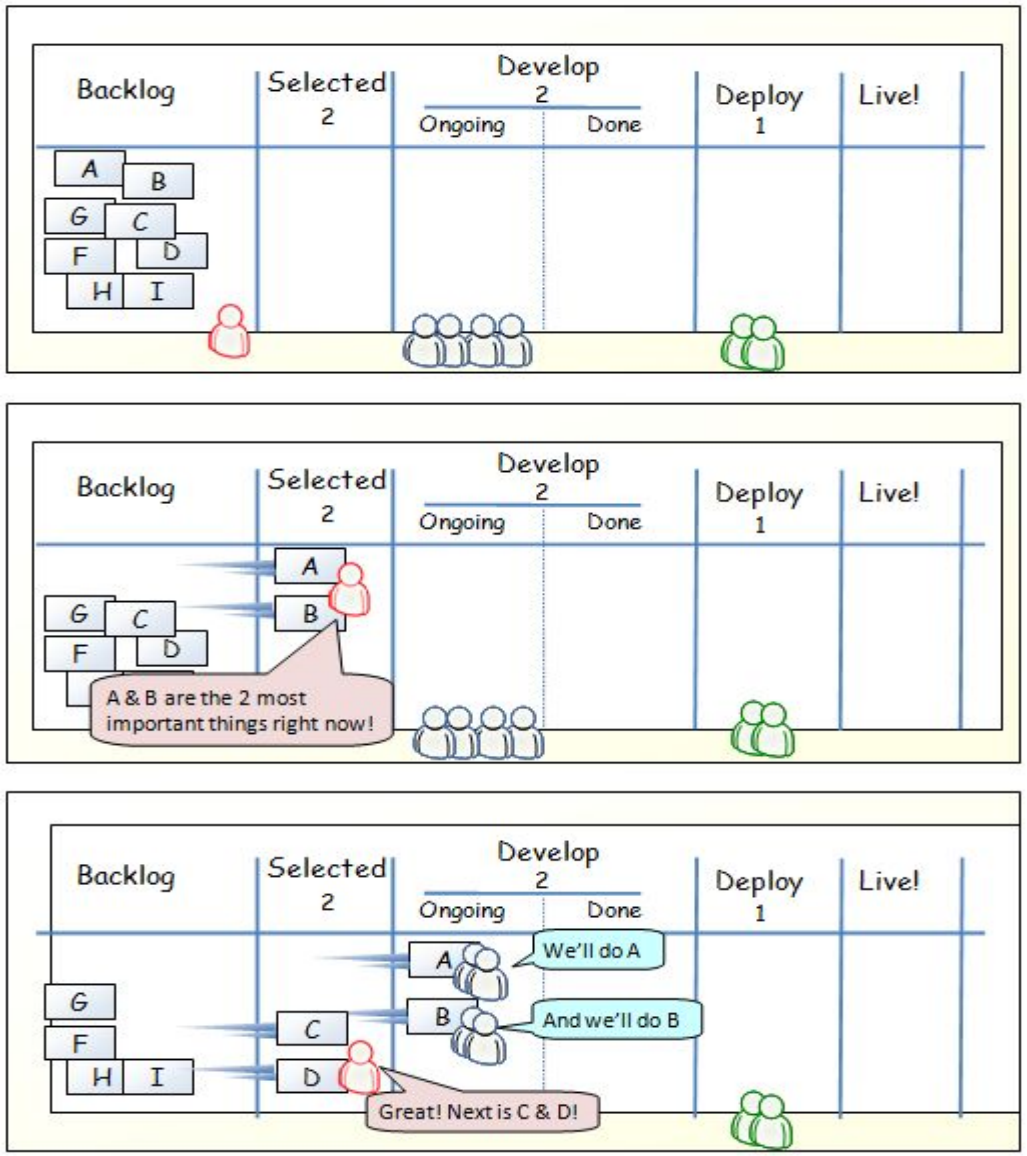
- Estimate ahead
  - Combination of average of last x iterations and team capacity
  - Rule of thumb: yesterday’s weather

Kanban in a nutshell

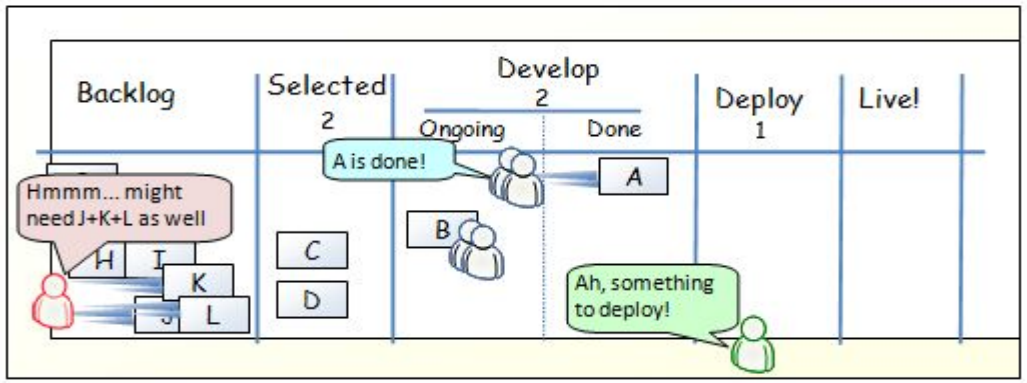
- Visualize the workflow
- Limit WIP (stop starting & start finishing)
- Measure & optimize flow

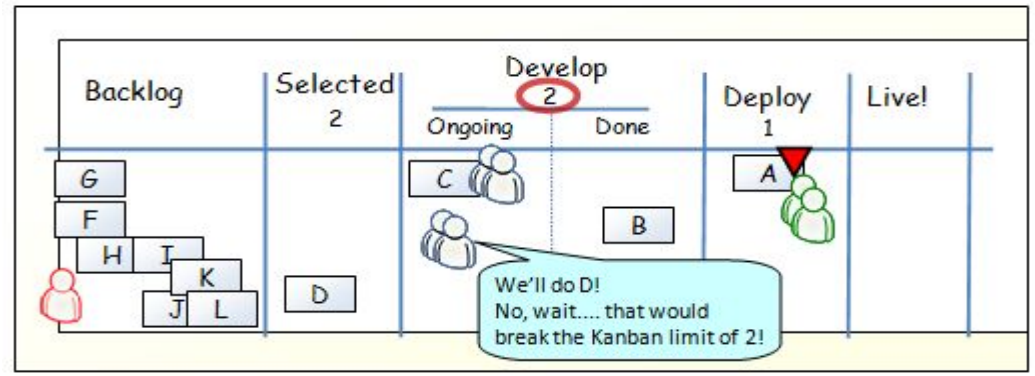


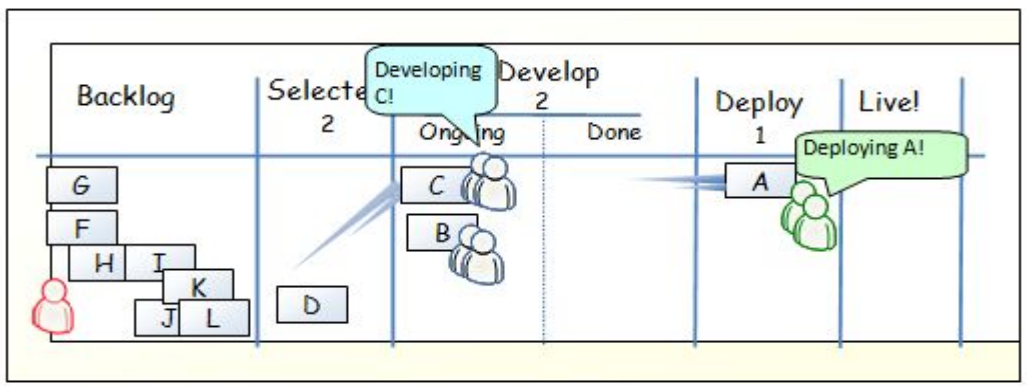
Example (Henrik Kniberg)

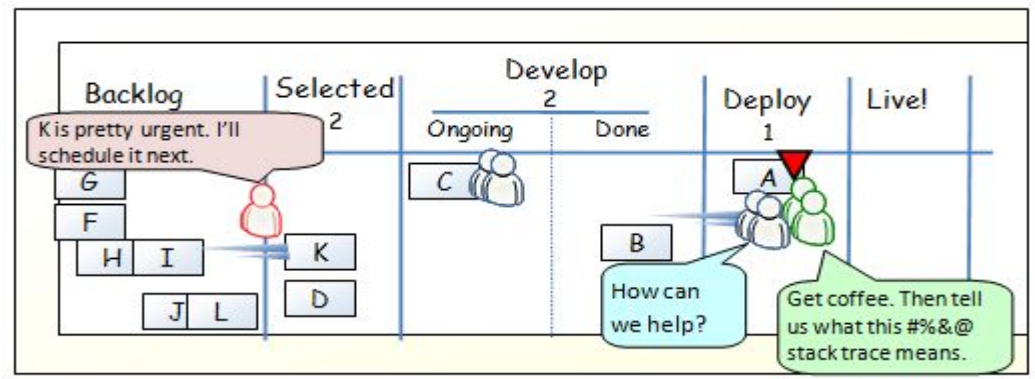


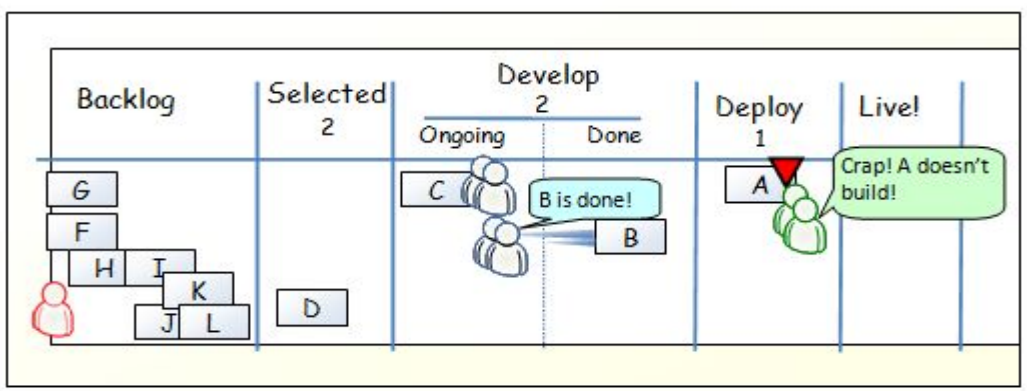
Example (continued)

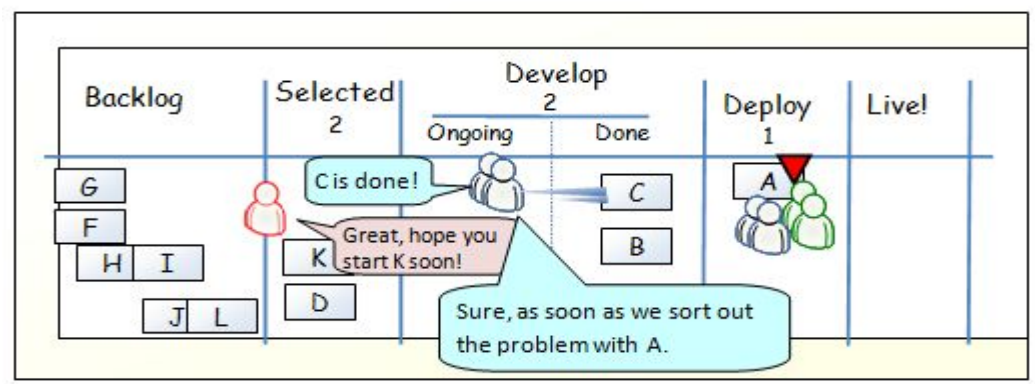












Prescriptive vs. Adaptive

More prescriptive

More adaptive

RUP (120+)

- Architecture Reviewer
- Business Designer
- Business-Model Reviewer
- Business-Process Analyst
- Capsule Designer
- Change Control Manager
- Code Reviewer
- Configuration Manager
- Course Developer
- Database Designer
- Deployment Manager
- Design Reviewer
- Designer
- Graphic Artist
- Implementer
- Integrator
- Process Engineer
- Project Manager
- Project Reviewer
- Requirements Reviewer
- Requirements Specifier
- Software Architect
- Stakeholder
- System Administrator
- System Analyst
- Technical Writer
- Test Analyst
- Test Designer
- Test Manager
- Tester
- Tool Specialist
- User-Interface Designer
- Architectural analysis
- Assess Viability of architectural proof-of-concept
- Capsule design
- Class design
- Construct architectural proof-of-concept
- Database design
- Describe distribution
- Describe the run-time architecture
- Design test packages and classes
- Develop design guidelines
- Develop programming guidelines
- Identify design elements
- Identify design mechanisms
- Incorporate design elements
- Prioritize use cases
- Review the architecture
- Review the design
- Structure the Implementation model
- Subsystem design
- Use-case analysis
- Use-case design
- Analysis model
- Architectural proof-of-concept
- Bill of materials
- Business architecture document
- Business case
- Business glossary
- Business modeling guidelines
- Business object model
- Business rules
- Business use case
- Business use case realization
- Business use-case model
- Business vision
- Change request
- Configuration audit findings
- Configuration management plan
- Data model
- Deployment model
- Deployment plan
- Design guidelines
- Design model
- Development case
- Development-organization assessment
- End-user support materials
- Glossary
- Implementation model
- Installation artifacts
- Integration build plan
- Issue list
- Iteration assessment
- Iteration plan
- Manual styleguide
- Programming guidelines
- Quality assurance plan
- Reference architecture
- Release notes
- Requirements attributes
- Requirements management plan
- Review record
- Risk list
- Risk management plan
- Software architecture document
- Software development plan
- Software requirements specification
- Stakeholder requests
- Status assessment
- Supplementary business specification
- Supplementary specification
- Target organization assessment
- Test automation architecture
- Test case
- Test environment configuration
- Test evaluation summary
- Test guidelines
- Test ideas list
- Test interface specification
- Test plan
- Test suite
- Tool guidelines
- Training materials
- Use-case model
- Use-case package
- Use-case modeling guidelines
- Use-case realization
- Use-case storyboard
- User-interface guidelines
- User-interface prototype
- Vision
- Work order
- Workload analysis model

XP (13)

- Whole team
- Coding standard
- TDD
- Collective ownership
- Continuous integration
- Pair programming
- Refactoring
- Planning game
- Continuous integration
- Simple design
- Sustainable pace
- Metaphor
- Small releases

Scrum (9)

- Scrum Master
- Product Owner
- Team
- Sprint planning meeting
- Daily Scrum
- Sprint review
- Product backlogt
- Sprint backlog
- Burndown chart

Kanban (3)

- Visualize the workflow
- Limit WIP
- Measure and optimize lead time

Do Whatever (0)

Do not develop an attachment
to any one weapon
or any one school of fighting

Miyamoto Musashi
17th century samurai

Kniberg

Example of a backlog management tool

- PivotalTracker: https://www.pivotaltracker.com/

  (you can sign up for free)

Two models, based on the examples discussed during the lecture:

- Parking Finder: https://www.pivotaltracker.com/n/projects/1978201
- RATB Finder: https://www.pivotaltracker.com/n/projects/1978171