Lecture 5: Intro To Process

Milestones, Estimation, Planning

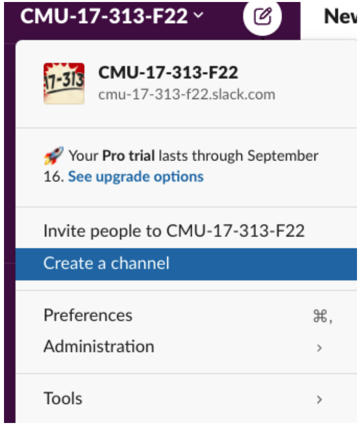
17-313 Fall 2022

Learning Goals

- Recognize the importance of process
- Understand the difficulty of measuring progress
- Identify why software development has project characteristics
- Use milestones for planning and progress measurement
- Understand backlogs and user stories
- Know your team!

Administrivia

- HW1 needs link to issue/PR. Please submit if not yet done.
- HW2 released on course website
  - Team assignment. Let us know if you don’t know your team!
  - Two deadlines (Sep 22nd and 27th) and three milestones
  - HIGHLY recommend completing first milestone this week
    - Create issue by Sep 15th to get the ball rolling
    - Soft deadline for getting feedback from staff
    - Fine to iterate/adapt through implementation (add comments to issue)
- Extra credit: Team activity
  - Create **private** channel on Slack
  - Invite your TA mentors to claim credit



CMU-17-313-F22

Nev

CMU-17-313-F22
cmu-17-313-f22.slack.com

🚀 Your **Pro trial** lasts through September 16. See upgrade options

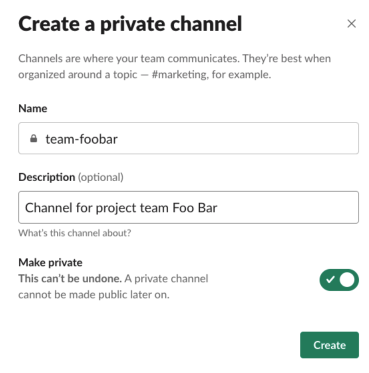
Invite people to CMU-17-313-F22

Create a channel

Preferences ⌘,

Administration ›

Tools ›



Create a private channel ×

Channels are where your team communicates. They’re best when organized around a topic — #marketing, for example.

Name

🔒 team-foobar

Description (optional)

Channel for project team Foo Bar

What’s this channel about?

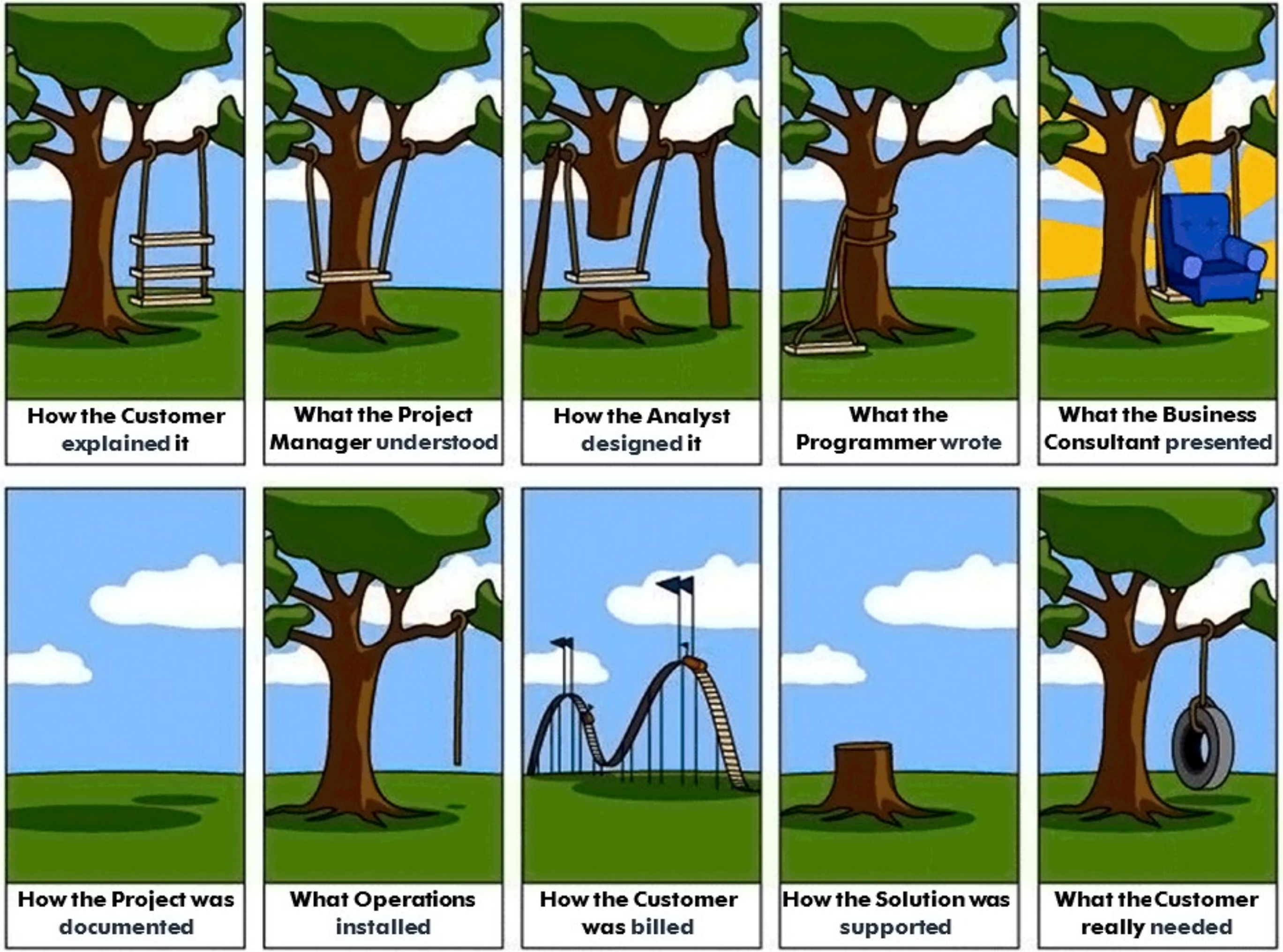
Make private
This can’t be undone. A private channel cannot be made public later on. ✓

Create

Software Process

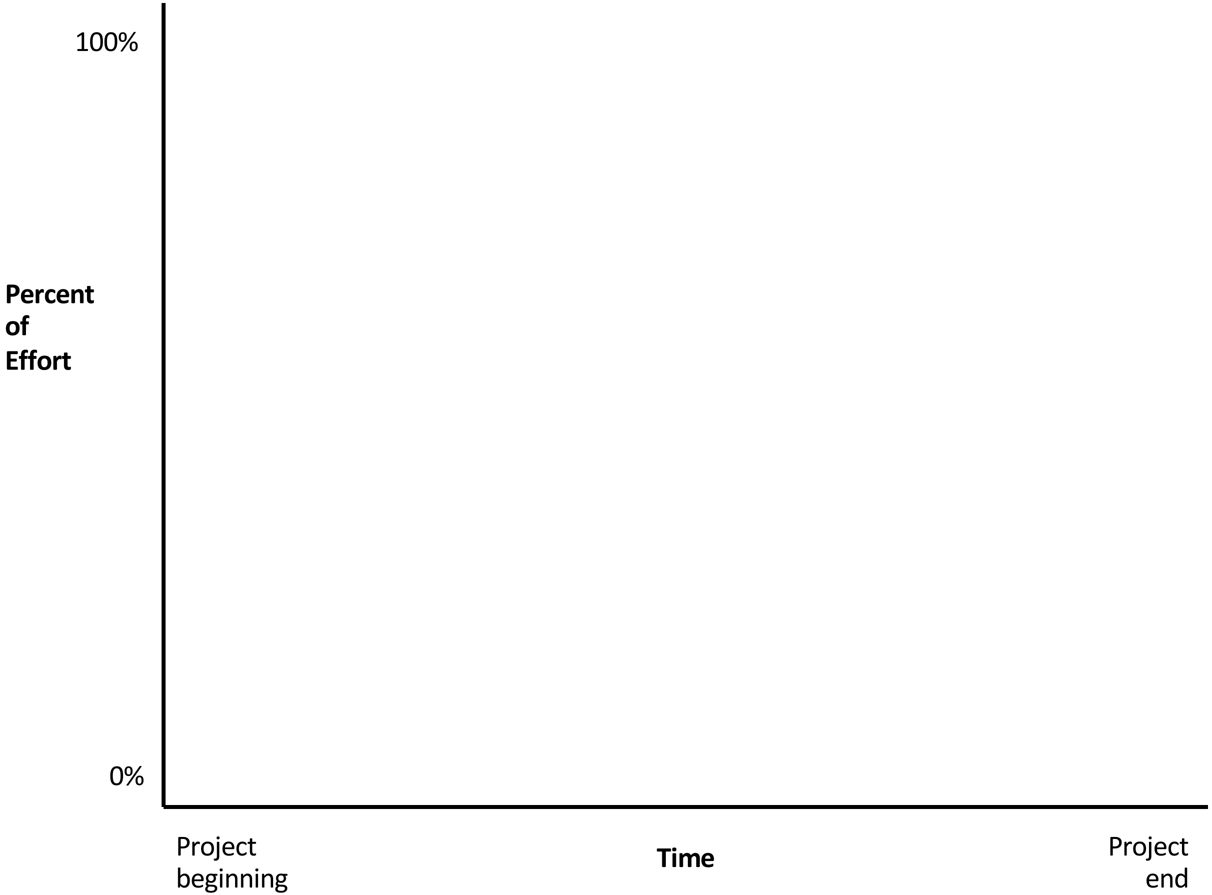
“The set of activities and associated results that produce a software product”

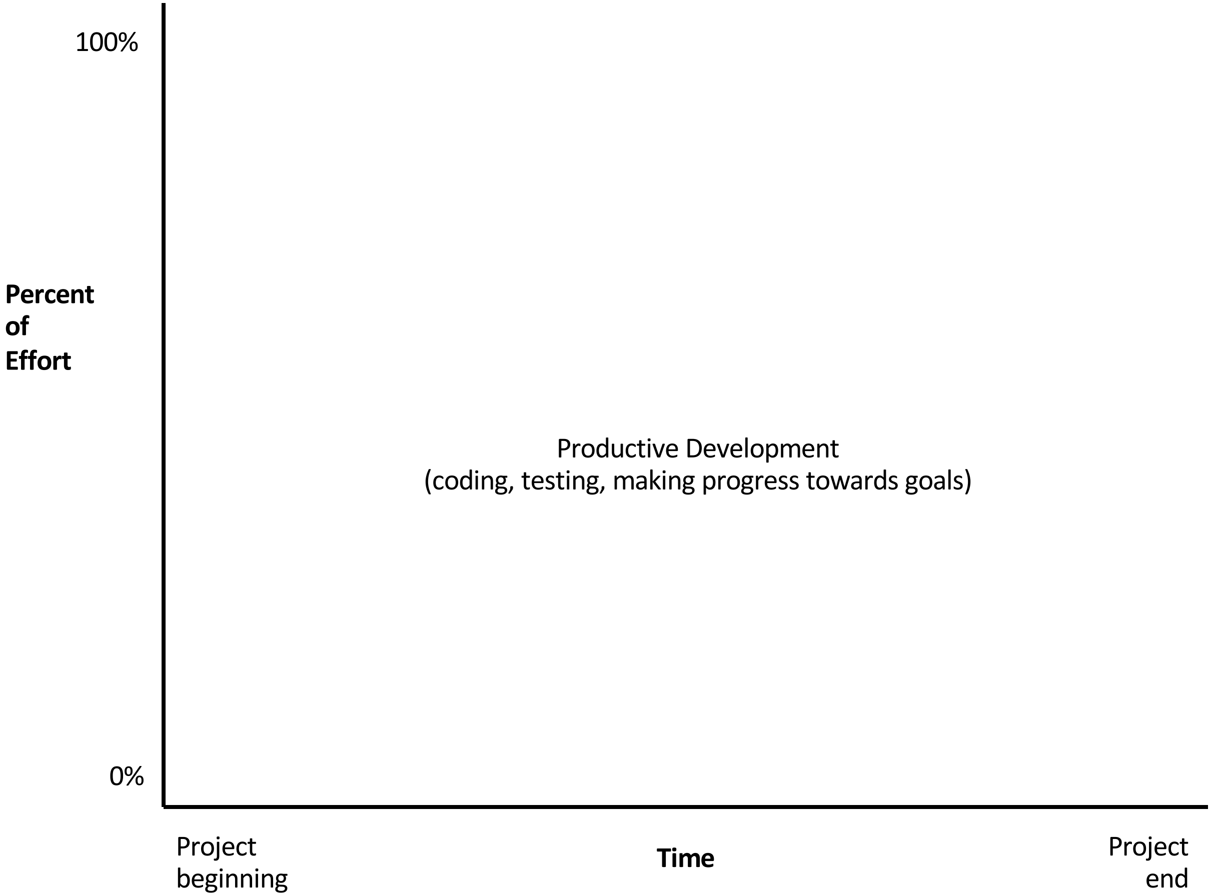
Sommerville, SE, ed. 8

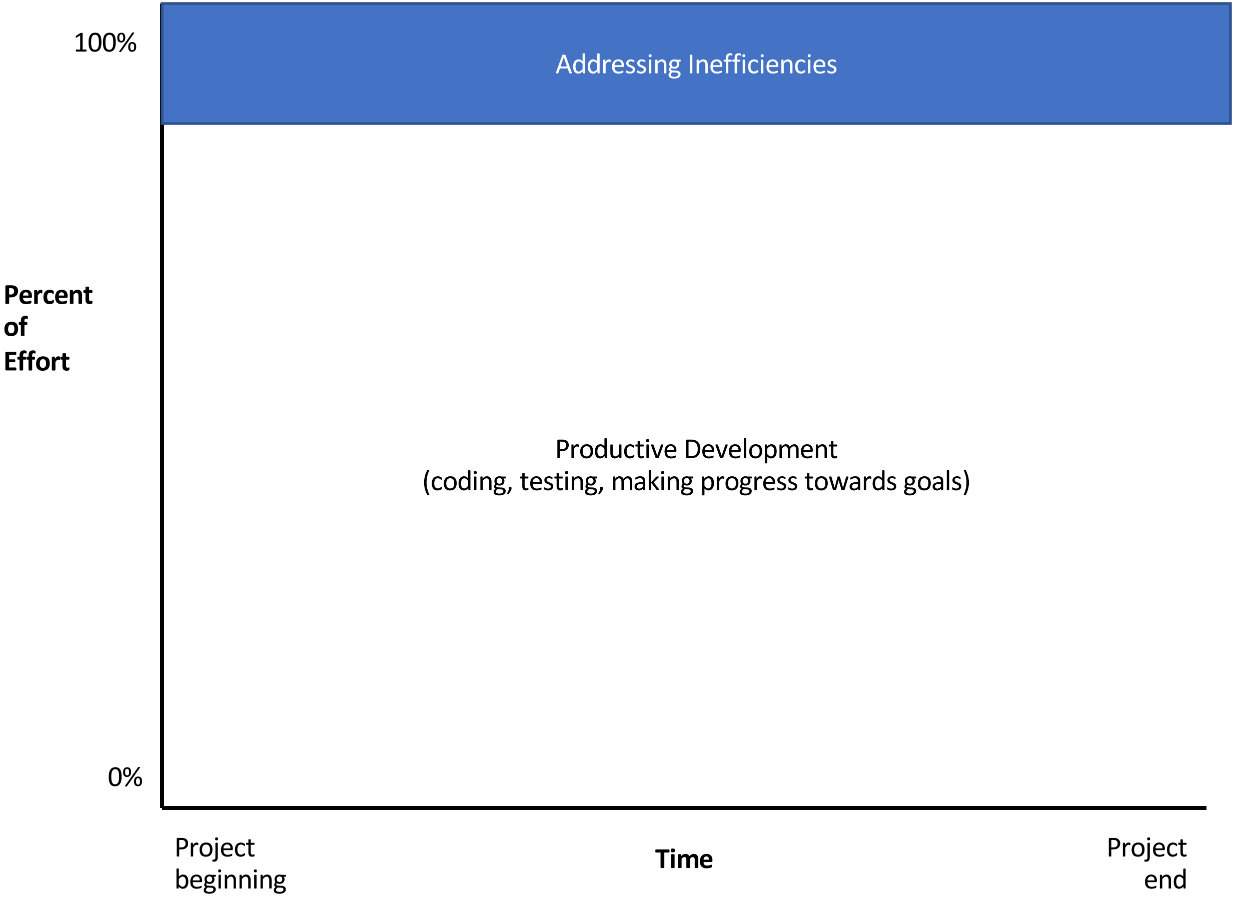
How the Customer explained it
What the Project Manager understood
How the Analyst designed it
What the Programmer wrote
What the Business Consultant presented
How the Project was documented
What Operations installed
How the Customer was billed
How the Solution was supported
What the Customer really needed

How to develop software?

1. Discuss the software that needs to be written
2. Write some code
3. Test the code to identify the defects
4. Debug to find causes of defects
5. Fix the defects
6. If not done, return to step 1


100%
Percent of Effort
0%
Project beginning
Time
Project end


100%
Percent of Effort
Productive Development
(coding, testing, making progress towards goals)
0%
Project beginning
Time
Project end



100%

Addressing Inefficiencies

Percent of Effort

Productive Development
(coding, testing, making progress towards goals)

0%

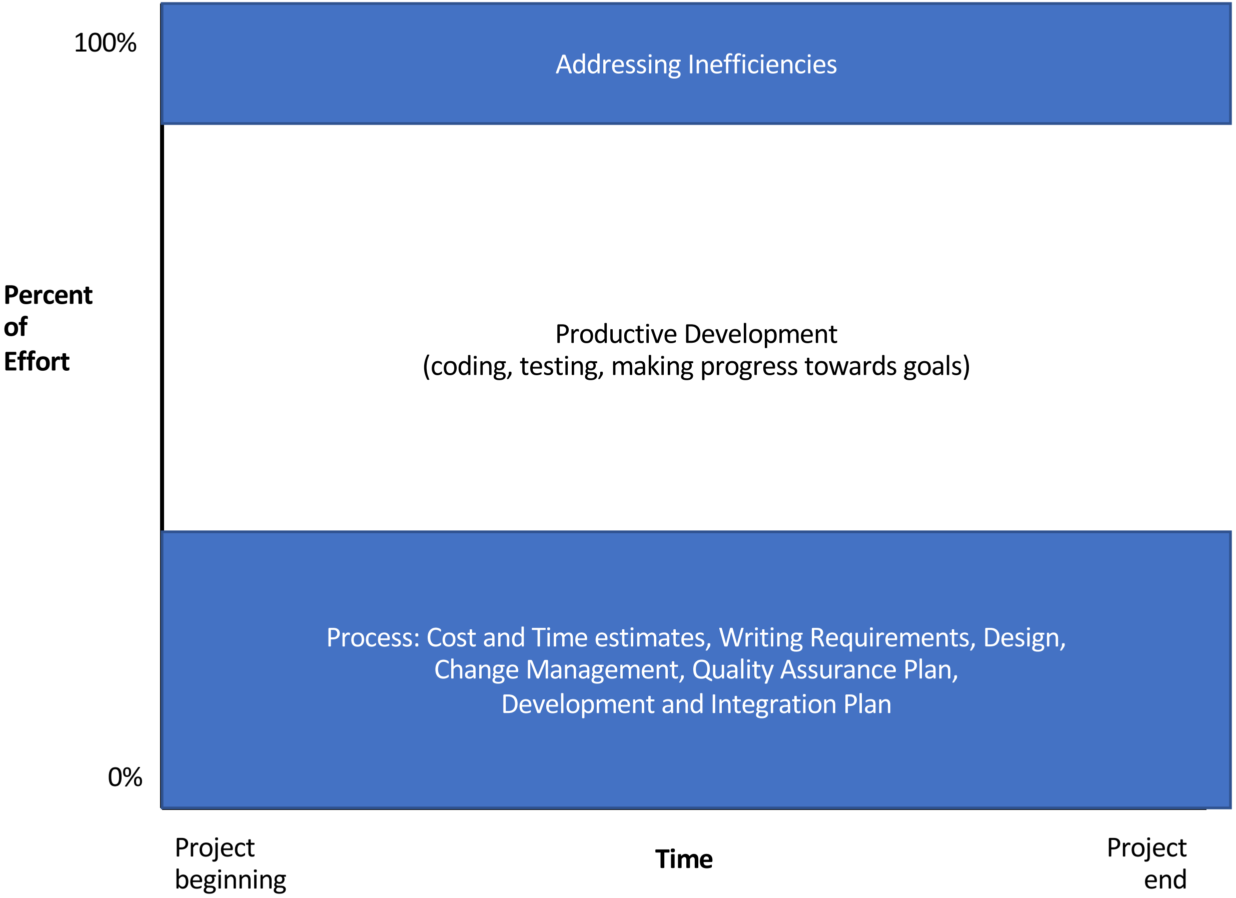
Project beginning

Time

Project end

Your manager asks you to follow a process

- Writing down all requirements
- Require approval for all changes to requirements
- Use version control for all changes
- Track all reported bugs
- Review requirements and code
- Break down development into smaller tasks and schedule and monitor them
- Planning and conducting quality assurance
- Have daily status meetings
- Use Docker containers to push code between developers and operation



100%

Addressing Inefficiencies

Percent of Effort

Productive Development
(coding, testing, making progress towards goals)

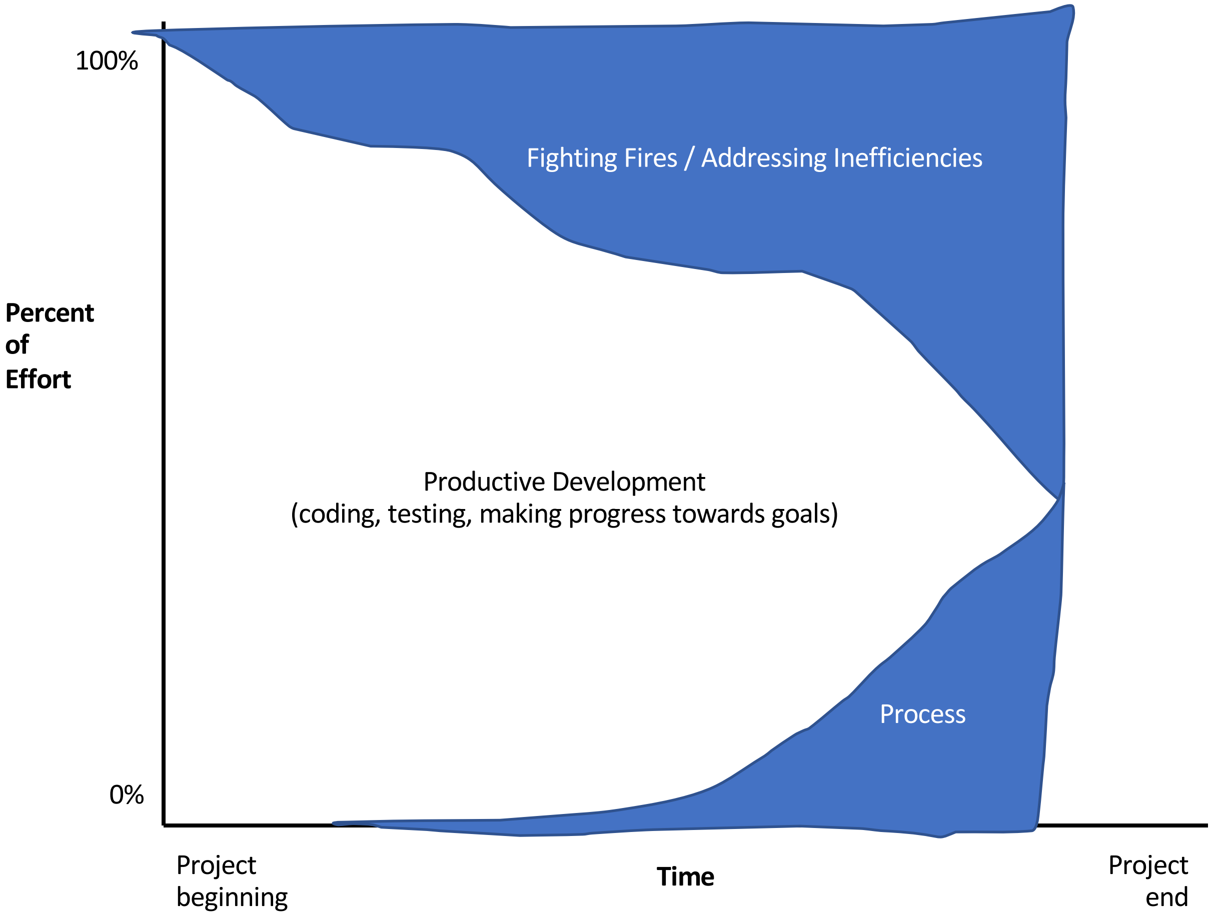
Process: Cost and Time estimates, Writing Requirements, Design, Change Management, Quality Assurance Plan, Development and Integration Plan

0%

Project beginning

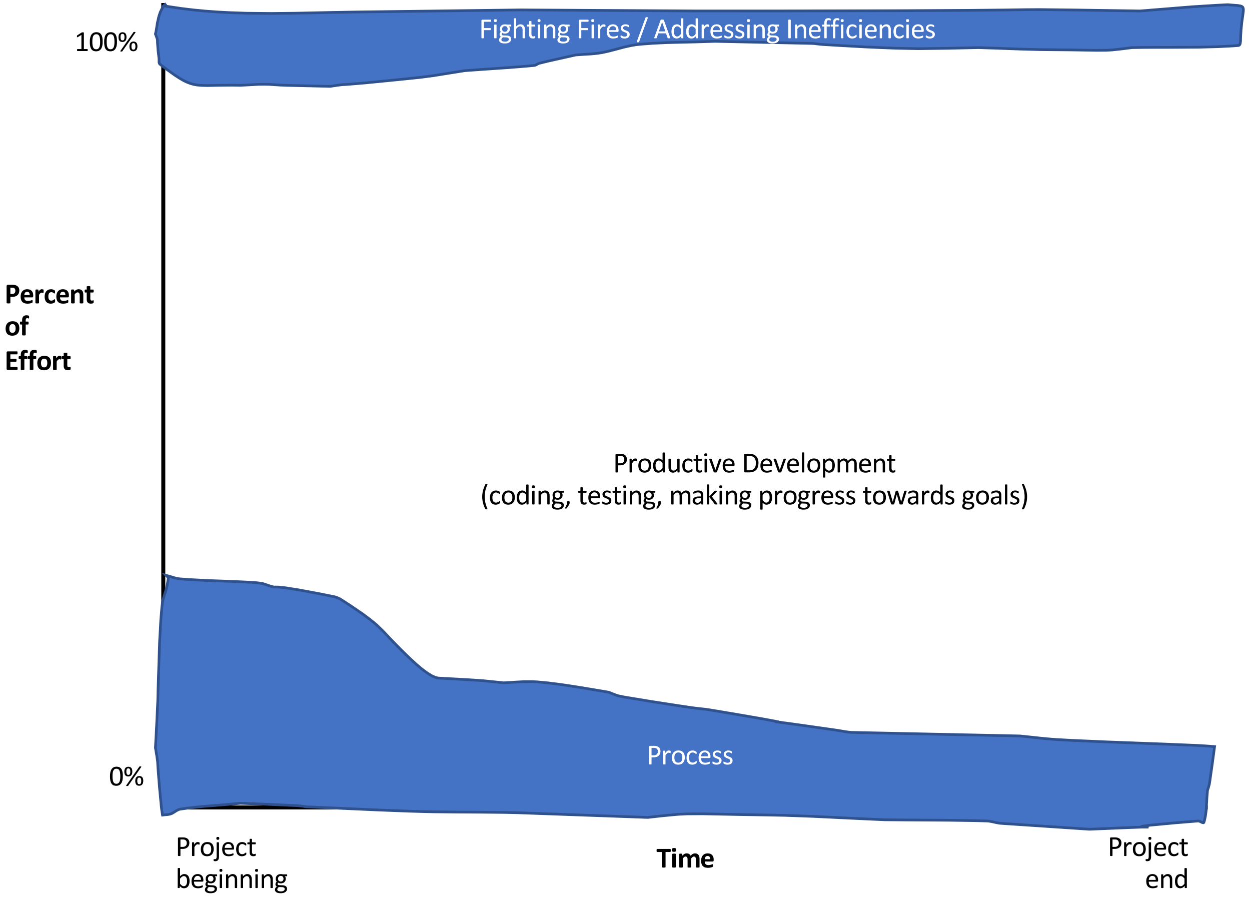
Time

Project end


100%
Percent
of
Effort
0%
Fighting Fires / Addressing Inefficiencies
Productive Development
(coding, testing, making progress towards goals)
Process
Project
beginning
Time
Project
end

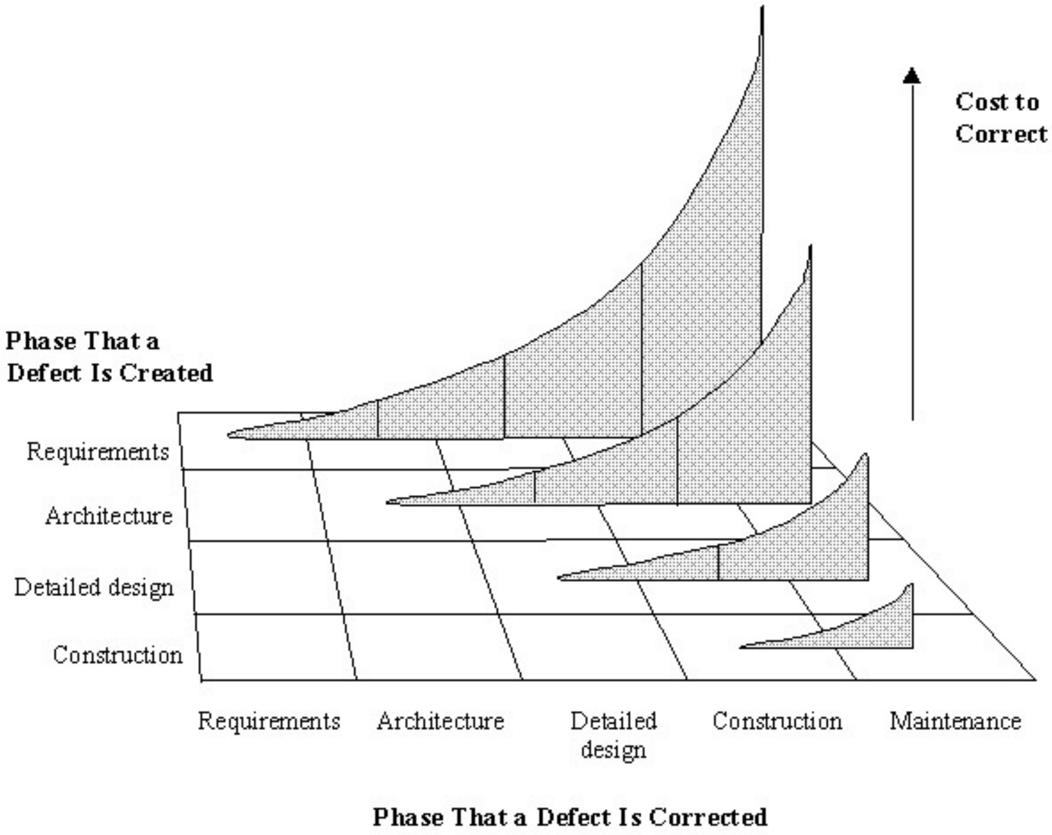
Example process issues

- Change Control: Mid-project informal agreement to changes suggested by customer or manager. Project scope expands 25-50%
- Quality Assurance: Late detection of requirements and design issues. Test-debug-reimplement cycle limits development of new features. Release with known defects.
- Defect Tracking: Bug reports collected informally, forgotten
- System Integration: Integration of independently developed components at the very end of the project. Interfaces out of sync.
- Source Code Control: Accidentally overwritten changes, lost work.
- Scheduling: When project is behind, developers are asked weekly for new estimates.



Hypothesis: Process increases flexibility and efficiency

Ideal Curve: Upfront investment for later greater returns


Cost to
Correct
Phase That a
Defect Is Created
Requirements
Architecture
Detailed design
Construction
Requirements
Architecture
Detailed
design
Construction
Maintenance
Phase That a Defect Is Corrected

Planning

Time estimation

I'M JUST OUTSIDE TOWN, SO I SHOULD BE THERE IN FIFTEEN MINUTES.
ACTUALLY, IT'S LOOKING MORE LIKE SIX DAYS.
NO, WAIT, THIRTY SECONDS.

THE AUTHOR OF THE WINDOWS FILE COPY DIALOG VISITS SOME FRIENDS.

https://xkcd.com/612/

Activity: Estimate Time

Task A: Simple web version of the Monopoly board game with Pittsburgh street names

Team: just you

Task B: Bank smartphone app

Team: you with team of 4 developers, one experienced with iPhone apps, one with background in security

* Estimate in 8h days (20 work days in a month, 220 per year)

My Task A estimate: ___
My Task B estimate: ___

Other Task A estimate: __
Other Task B estimate: __

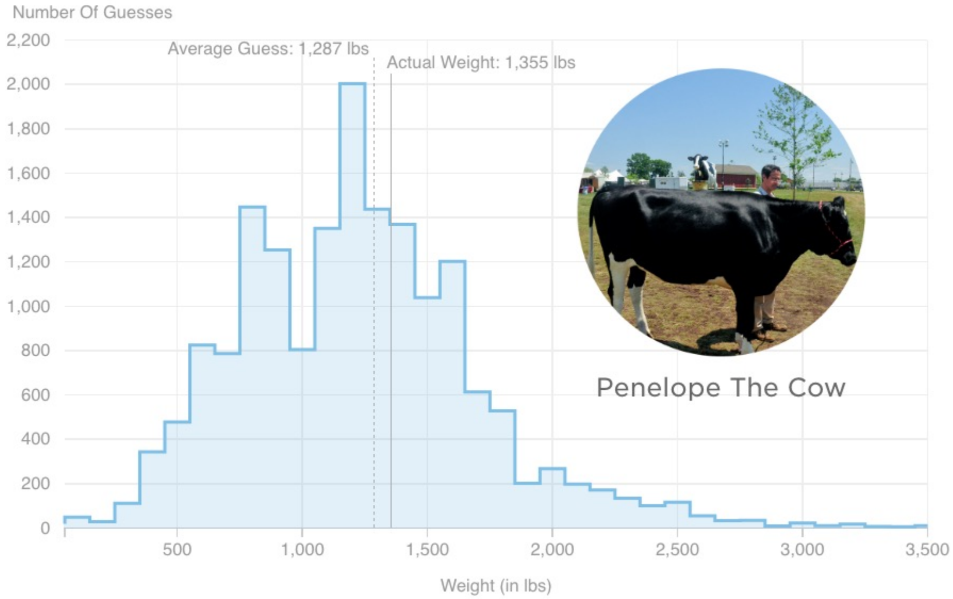
Other Task A estimate: __
Other Task B estimate: __

Revise Time Estimate

- Do you have comparable experience to base an estimate off of?
- How much design do you need for each task?
- Break down the task into ~5 smaller tasks and estimate them.
- Revise your overall estimate if necessary

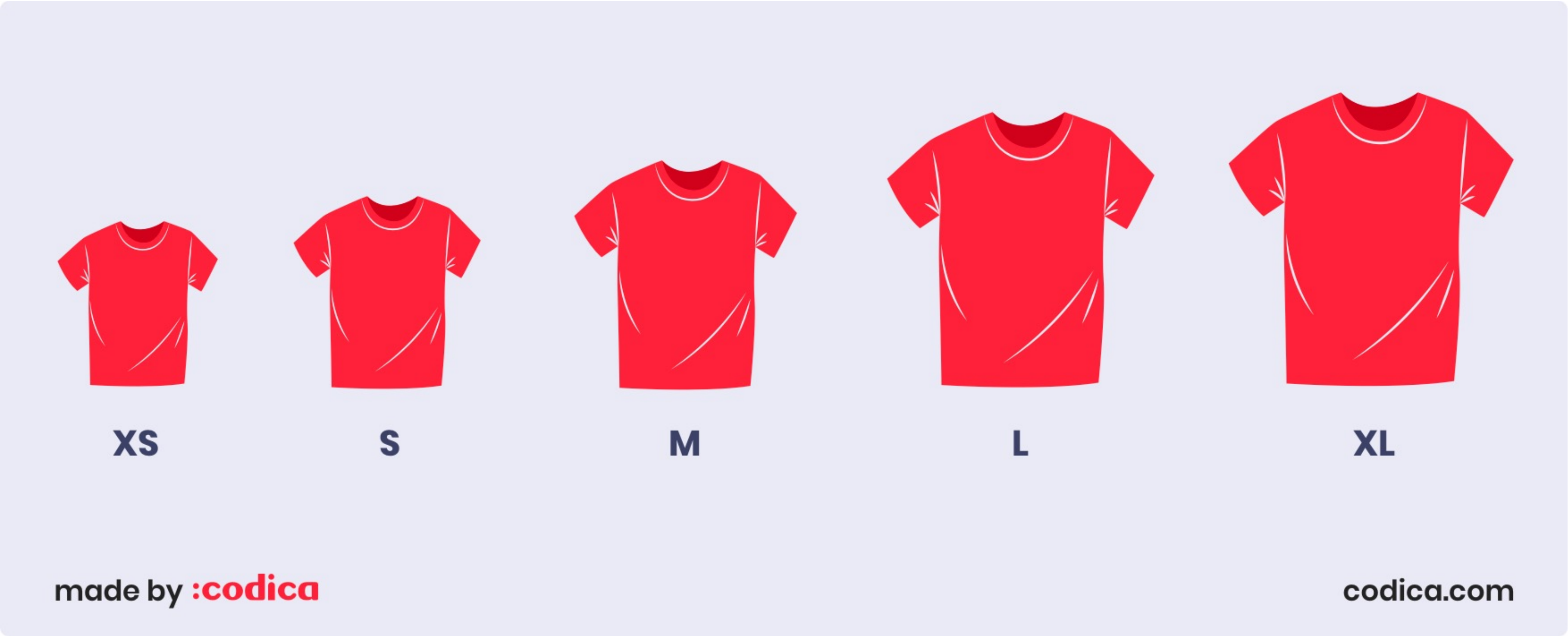
How Much Does This Cow Weigh?

(All People)



Source: The Internet.

Credit: Quoctrung Bui/NPR


XS
S
M
L
XL
made by :codica
codica.com

π

Measuring Progress?

- “I’m almost done with the app. The frontend is almost fully implemented. The backend is fully finished except for the one stupid bug that keeps crashing the server. I only need to find the one stupid bug, but that can probably be done in an afternoon. We should be ready to release next week.”

Measuring Progress?

- Developer judgment: x% done
- Lines of code?
- Functionality?
- Quality?

Milestones and deliverables make progress *observable*

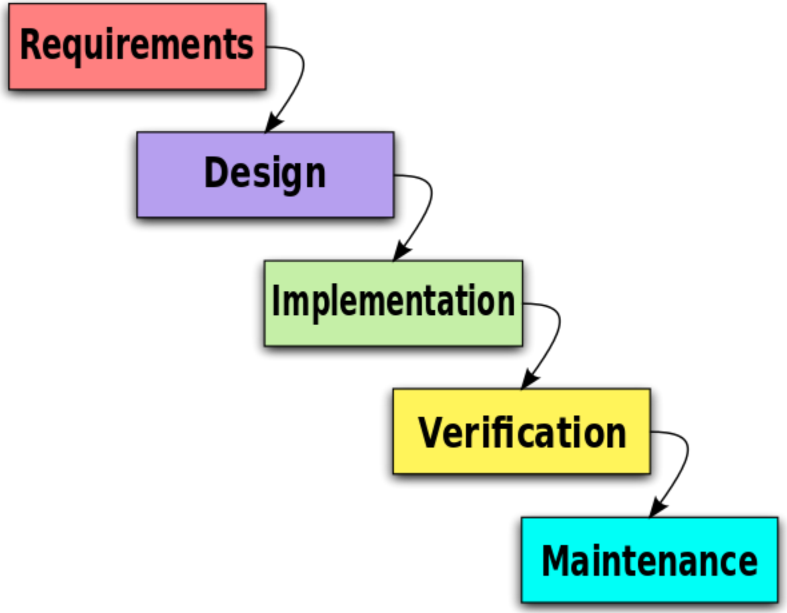
Milestone: clear end point of a (sub)tasks

- For project manager
- Reports, prototypes, completed subprojects
- "80% done" not a suitable mile stone

Deliverable: Result for customer

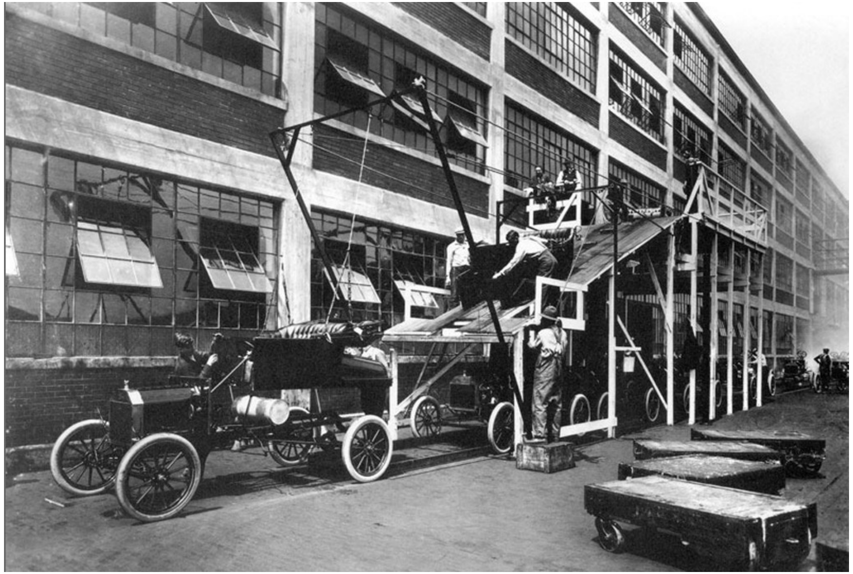
- Similar to milestone, but for customers
- Reports, prototypes, completed subsystems

Waterfall model was the original software process



Waterfall diagram CC-BY 3.0 **Paulsmith99 at en.wikipedia**

… akin to processes pioneered in mass manufacturing (e.g., by Ford)



Lean production adapts to variable demand

Taiichi Ohno

Toyota Production System (TPS)

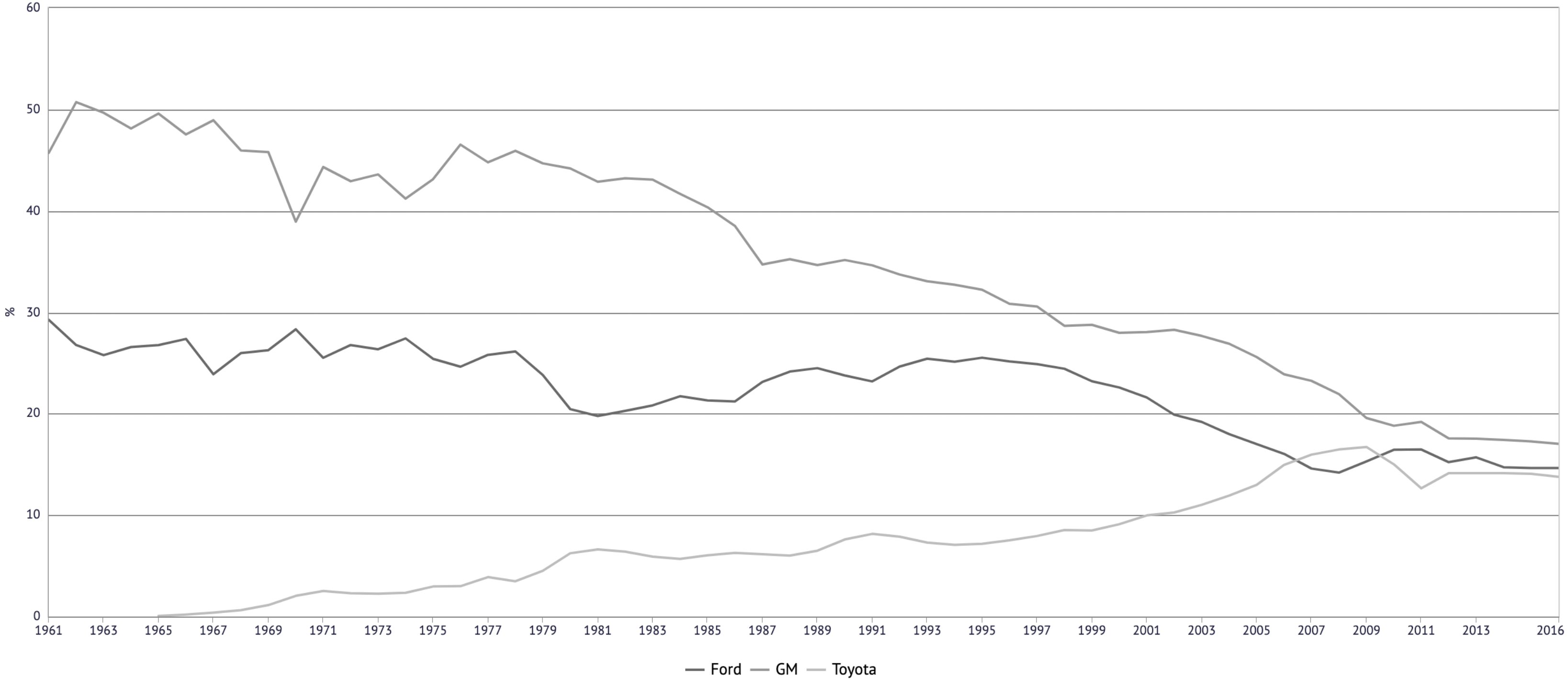
- Build only what is needed, only when it is needed.
- Use the “pull” system to avoid overproduction. (Kanban)
- Stop to fix problems, to get quality right from the start (Jidoka)
- Workers are multi-skilled and understand the whole process; take ownership

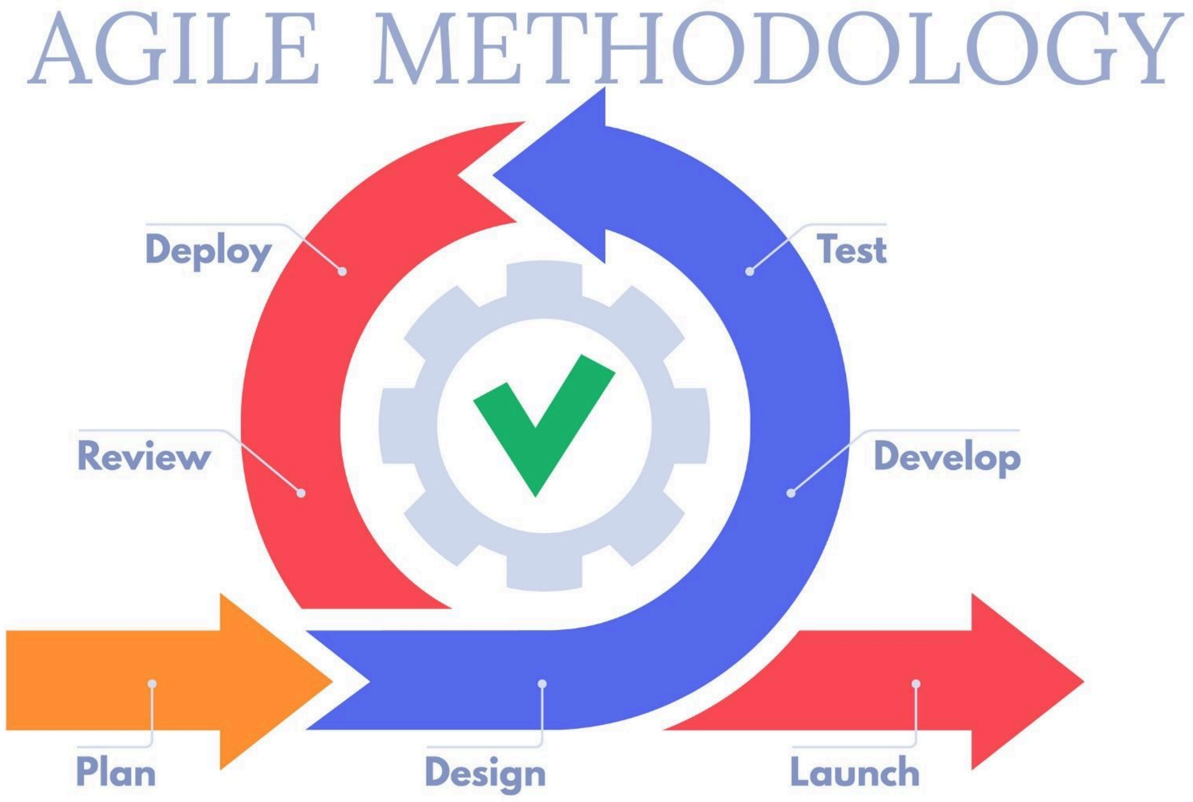
Lots of software buzzwords invented recently build on these ideas

- Just-in-time, DevOps, Shift-Left

See also: “The machine that changed the world” by James P Womack et al. The Free Press, 2007.

US vehicle sales market share; 1961—2016 (source: knoema.com)

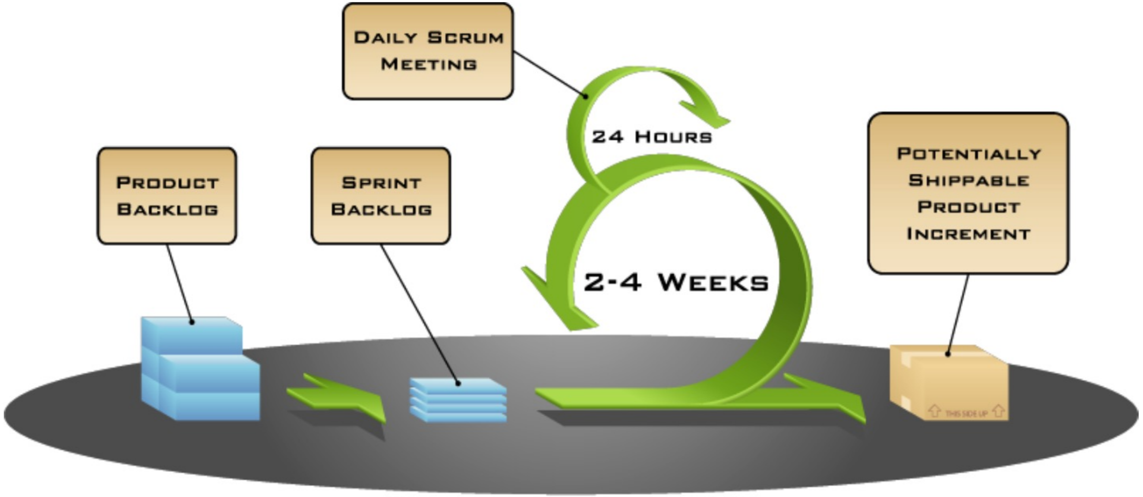



AGILE METHODOLOGY
Deploy
Test
Review
Develop
Plan
Design
Launch

Scrum

(Only a brief intro)

Elements of Scrum



Products:

- Product Backlog
- Sprint Backlog

Process:

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

Backlogs

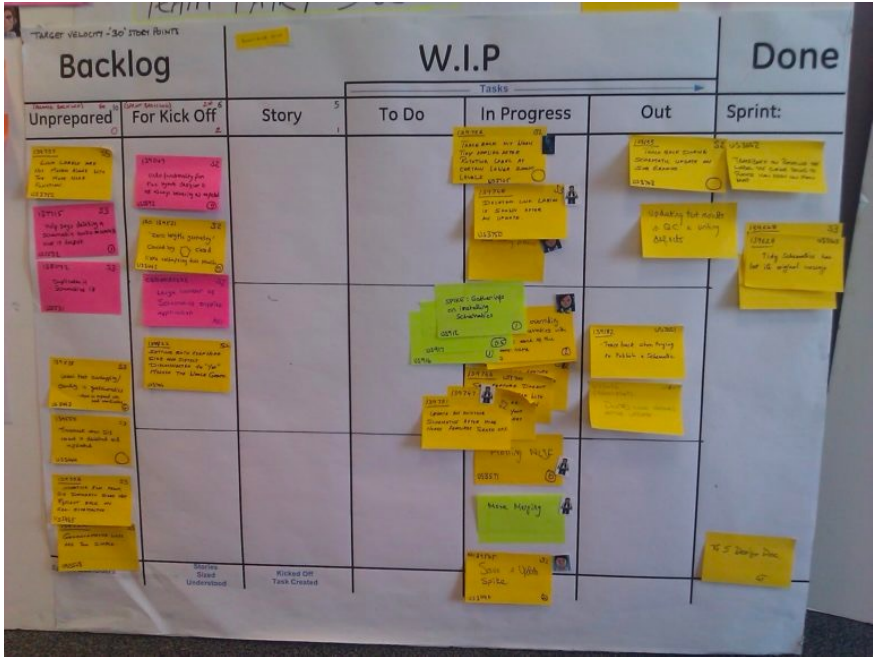
The **product backlog** is all the features for the product

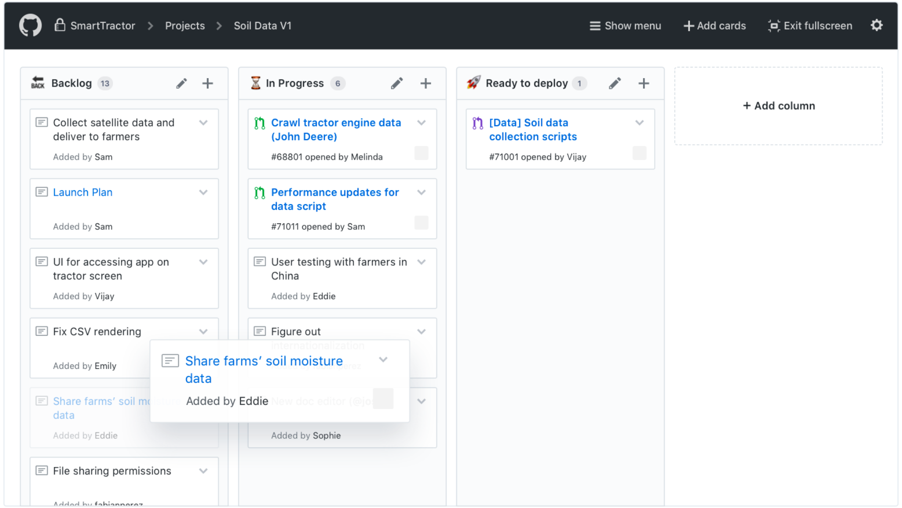
The **sprint backlog** is all the features that will be worked on for that sprint. These should be broken down into discrete tasks:

- Fine-grained
- Estimated
- Assigned to individual team members
- Acceptance criteria should be defined

User Stories are often used

Kanban boards





Backlog 13

- Collect satellite data and deliver to farmers
  Added by Sam
- Launch Plan
  Added by Sam
- UI for accessing app on tractor screen
  Added by Vijay
- Fix CSV rendering
  Added by Emily
- Share farms' soil moisture data
  Added by Eddie
- File sharing permissions
  Added by fabianperez

In Progress 6

- Crawl tractor engine data (John Deere)
  #68801 opened by Melinda
- Performance updates for data script
  #71011 opened by Sam
- User testing with farmers in China
  Added by Eddie
- Figure out internationalization
- New doc editor (@jo
  Added by Sophie

Ready to deploy 1

- [Data] Soil data collection scripts
  #71001 opened by Vijay

Share farms' soil moisture data
Added by Eddie

Scrum Meetings

Sprint Planning Meeting

- Entire Team decides together what to tackle for that sprint

Daily Scrum Meeting

- Quick Meeting to touch base on :
  - What have I done? What am I doing next? What am I stuck on/need help?

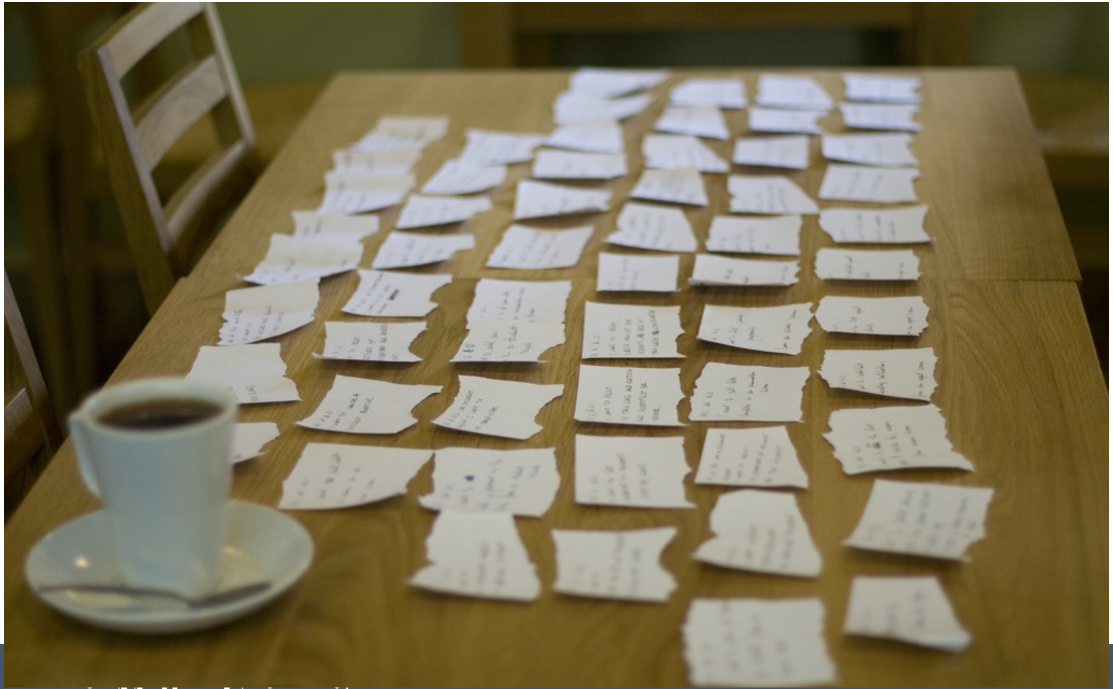
Sprint Retrospective

- Review sprint process

Sprint Review Meeting

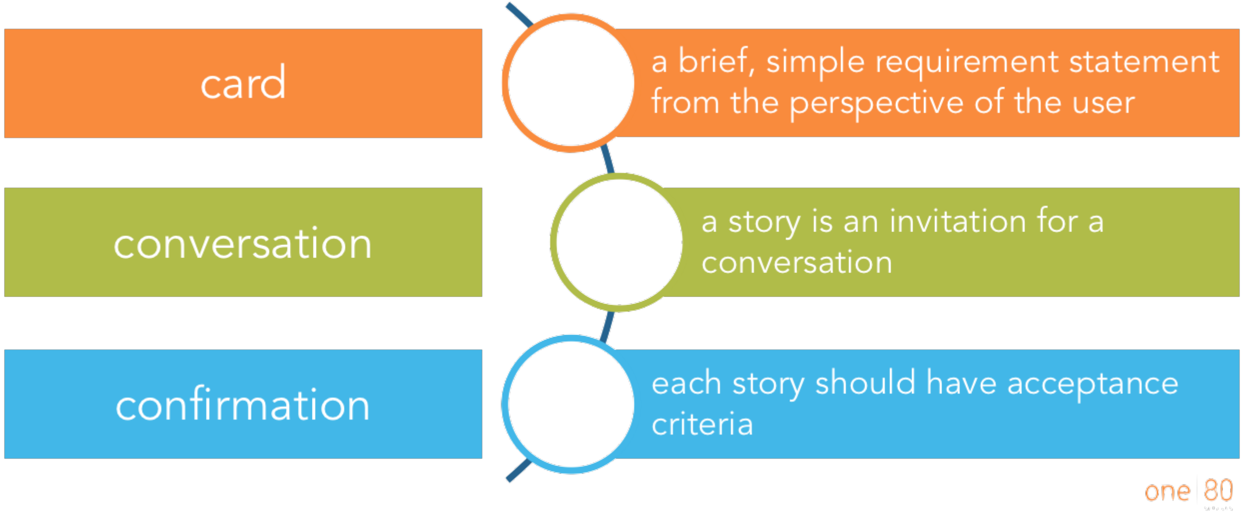
- Review Product

User Stories



Source: https://www.flickr.com/photos/jakuza/2728096478

User Stories



card

a brief, simple requirement statement from the perspective of the user

conversation

a story is an invitation for a conversation

confirmation

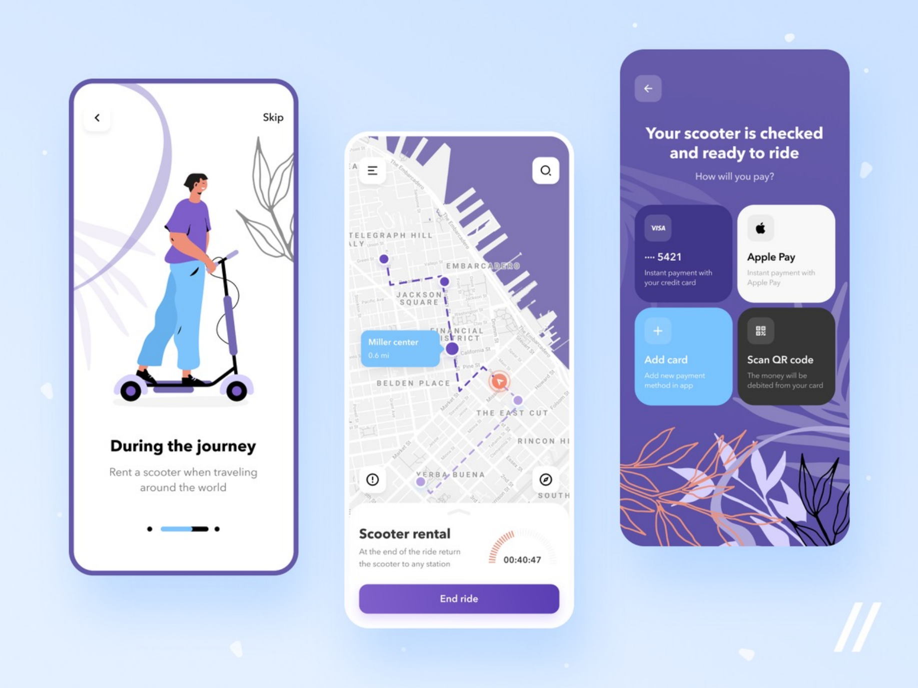
each story should have acceptance criteria

Source: http://one80services.com/user-stories/writing-good-user-stories-hint-its-not-about-writing/

User story cards (3”x5”)

“As a [role], I want [function], so that [value]”

Exercise



During the journey

Rent a scooter when traveling around the world

Scooter rental

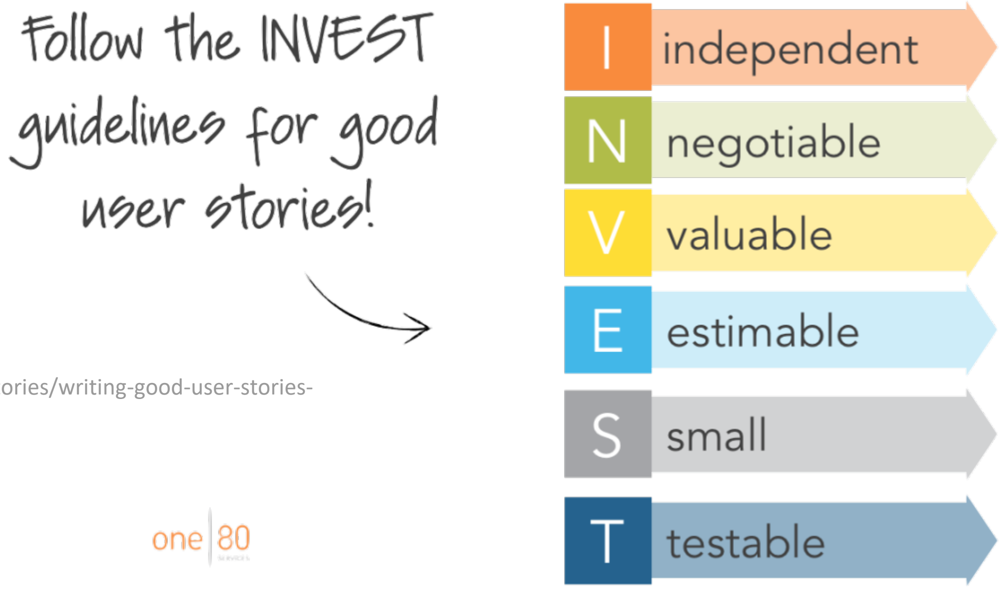
At the end of the ride return the scooter to any station

Your scooter is checked and ready to ride

How will you pay?

https://dribbble.com/shots/12512417-Scooter-Rental-App-Design

How to evaluate user story?



Follow the INVEST guidelines for good user stories!

I independent

N negotiable

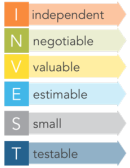
V valuable

E estimable

S small

T testable

Source: http://one80services.com/user-stories/writing-good-user-stories-hint-its-not-about-writing/



I independent

N negotiable

V valuable

E estimable

S small

T testable

Independent

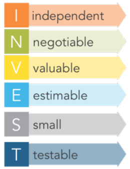
- Schedule in any order.
- Not overlapping in concept
- Not always possible

Negotiable

I independent
N negotiable
V valuable
E estimable
S small
T testable

- Details to be negotiated during development
- Good Story captures the essence, not the details

Valuable


I independent
N negotiable
V valuable
E estimable
S small
T testable

- This story needs to have value to someone (hopefully the customer)
- Especially relevant to splitting up issues

Estimable

I independent
N negotiable
V valuable
E estimable
S small
T testable

- Helps keep the size small
- Ensure we negotiated correctly
- “Plans are nothing, planning is everything” -Dwight D. Eisenhower

Small

I independent

N negotiable

V valuable

E estimable

S small

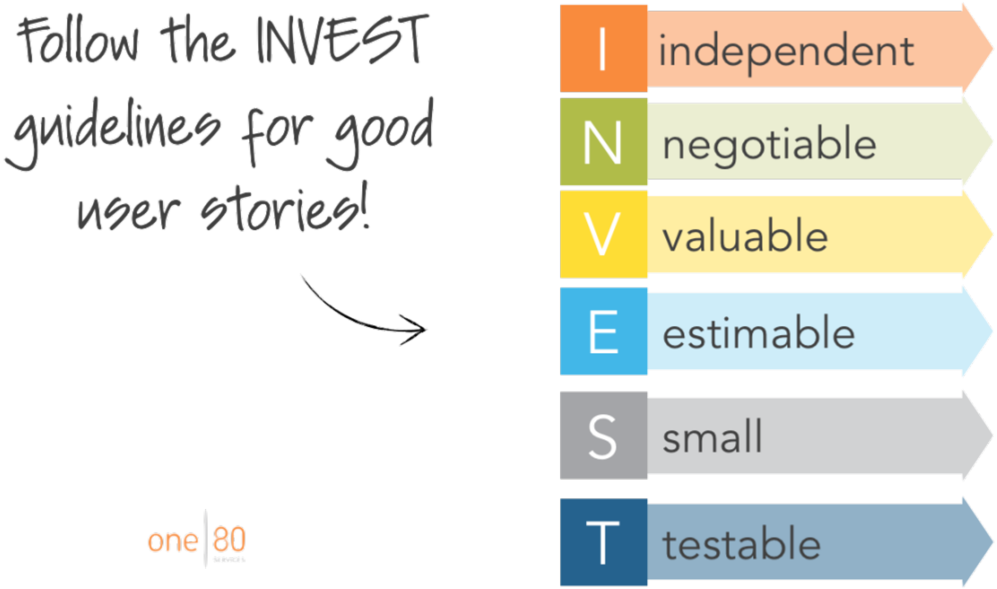
T testable

- Fit on 3x5 card
- At most two person-weeks of work
- Too big == unable to estimate

Testable

- Ensures understanding of task
- We know when we can mark task “Done”
- Unable to test == do not understand

Activity



Follow the INVEST guidelines for good user stories!

- I independent
- N negotiable
- V valuable
- E estimable
- S small
- T testable

Next up: Teams