Intro To Process: Milestones, Estimation, Planning

17-313: Foundations of Software Engineering

https://cmu-313.github.io

Michael Hilton and Chris Timperley

Fall 2025

Smoking Section

- Last full row

Administriva

- Some teams might have to be re-balanced, we will be in touch
- Extra Credit: go out for an activity
- NOTE ABOUT TEAMS: We might need to rebalance
- If you added late, see us about extension till TONIGHT

Project 1: Retrospective

P1: Retrospective (1/3)

- You had to **fork, build, and test** an unseen codebase 🏗️
  - gained experience with lots of tools you will use throughout the semester
  - had to work some tricky setup issues
  - you should now have a good foundation for P2

- You had to **change the code** to remove a code smell 🧀
  - some changes were harder to make than others
    - API changes → need to change all API uses

  - did you use AI to help you fix the smell? *(this was OK!)*

P1: Retrospective (2/3)

- Verifying those changes **is much harder!** ✅
  - did you break something? are you confident?
  - what assurances do you have?

- You used **code archaeology** to find out how to exercise them ▯
  - added probes to find out what code is being executed (e.g., console.log)
  - used code search to work backwards from code to frontend HTML
  - some issues were very difficult to trigger! did you swap to another one?

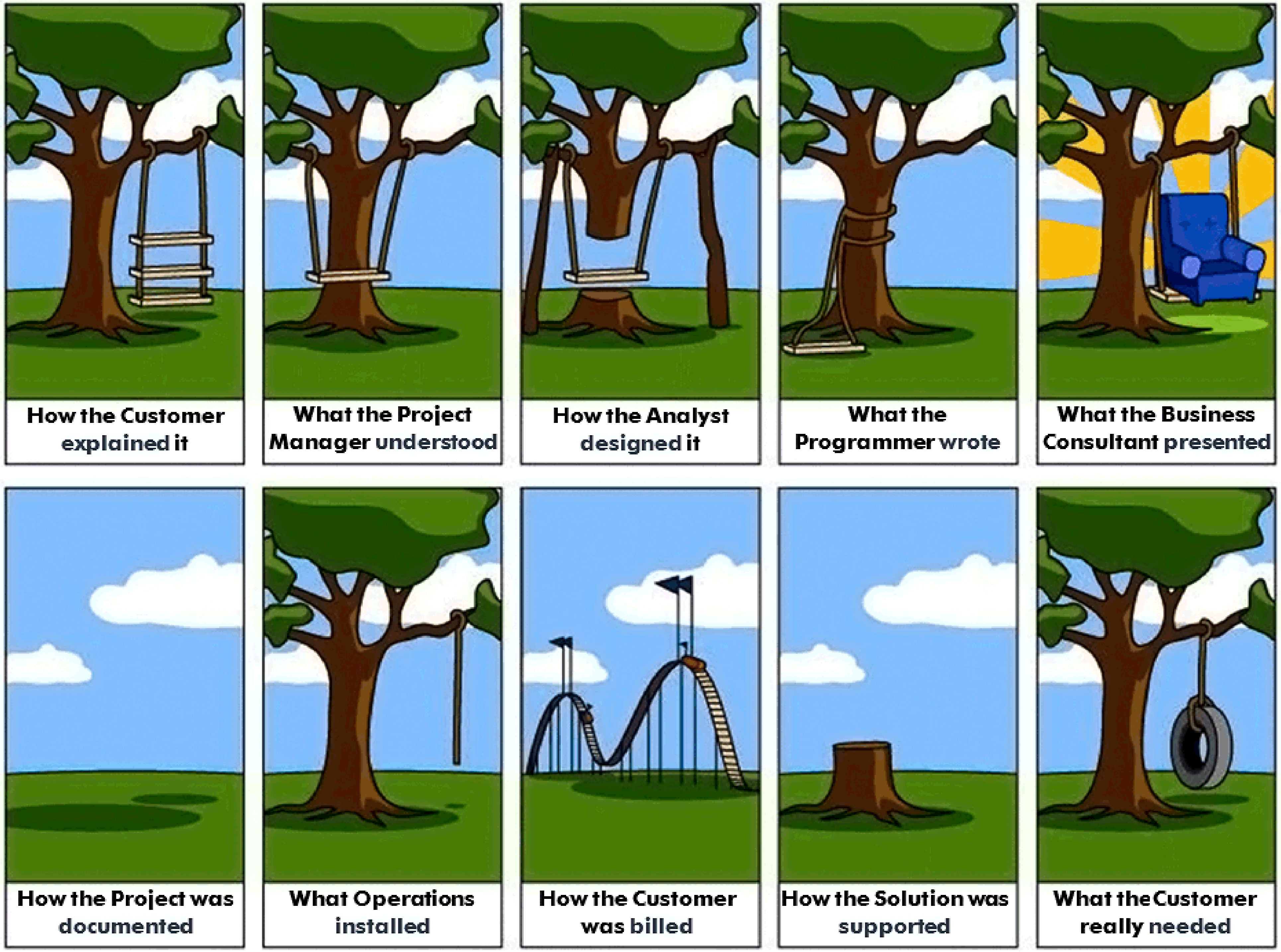
P1: Retrospective (3/3)

- You fixed the issue, but **did you really** fix the issue? 🤷‍♂️
  - did you **introduce more problems** into the code? extraneous changes?
  - **should you** have fixed the issue? was it even an issue to begin with?
    - did you make changes to appease the linter?

- If I merge every PR, **would the codebase be better?** 🤔
  - how do I review these PRs?
  - how do I make PRs that are more likely to be accepted?
  - is there a better way to confidently make large changes?

Today's Learning Goals

- Recognize the importance of process
- Identify why software development has project characteristics
- Understand the elements of Scrum
- Create and evaluate user stories
- Use milestones for planning and progress measurement
- Understand the difficulty of measuring progress

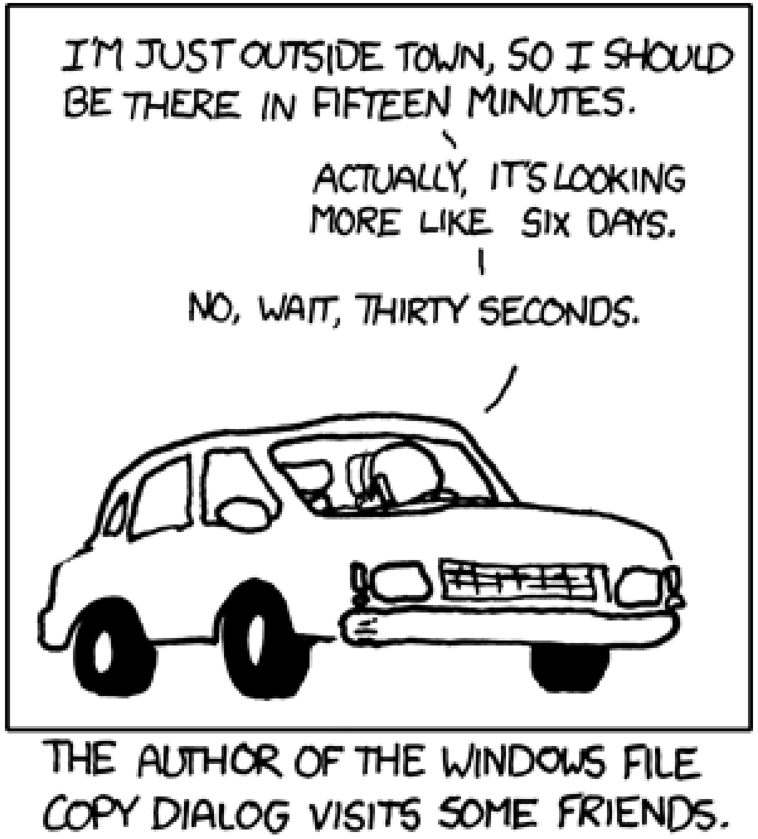

How the Customer explained it
What the Project Manager understood
How the Analyst designed it
What the Programmer wrote
What the Business Consultant presented
How the Project was documented
What Operations installed
How the Customer was billed
How the Solution was supported
What the Customer really needed

Software Process

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

Sommerville, SE, ed. 8

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: iOS app of the Monopoly board game with Pittsburgh street names with online play

Developer Team: just you

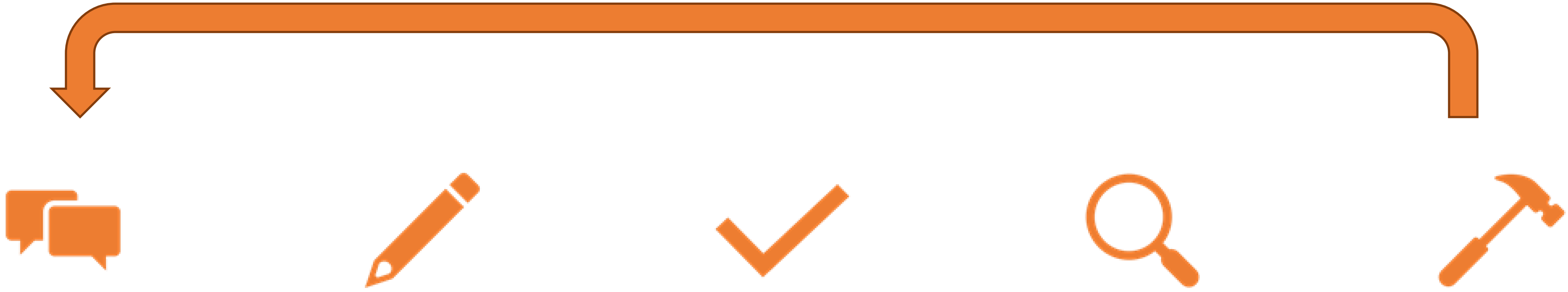
Justify your estimates

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

What does this mean?
What else can we do apart from coding?
Processes are **key** concerns.

Software En Principles, practices (technical and non-technical) for confidently building high-quality software.

How to develop software???



Discuss the software that needs to be written

Write some code

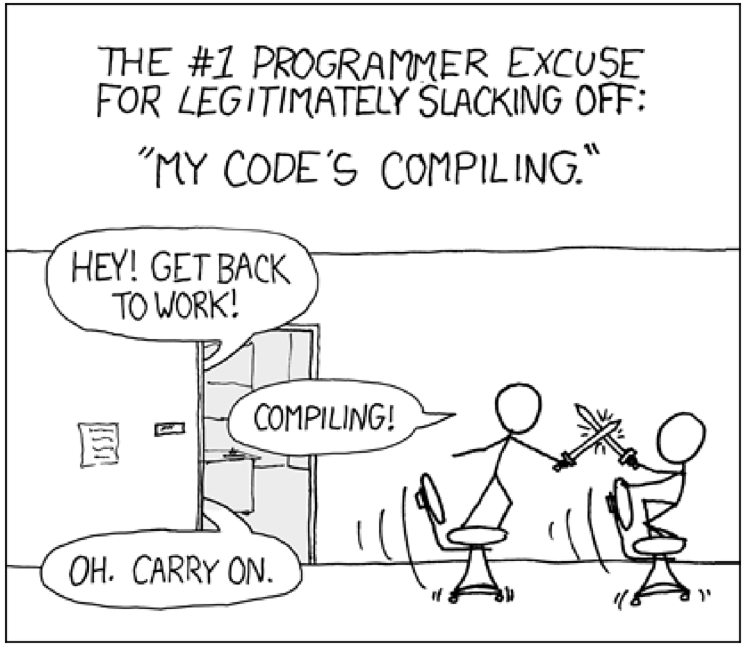
Test the code to identify the defects

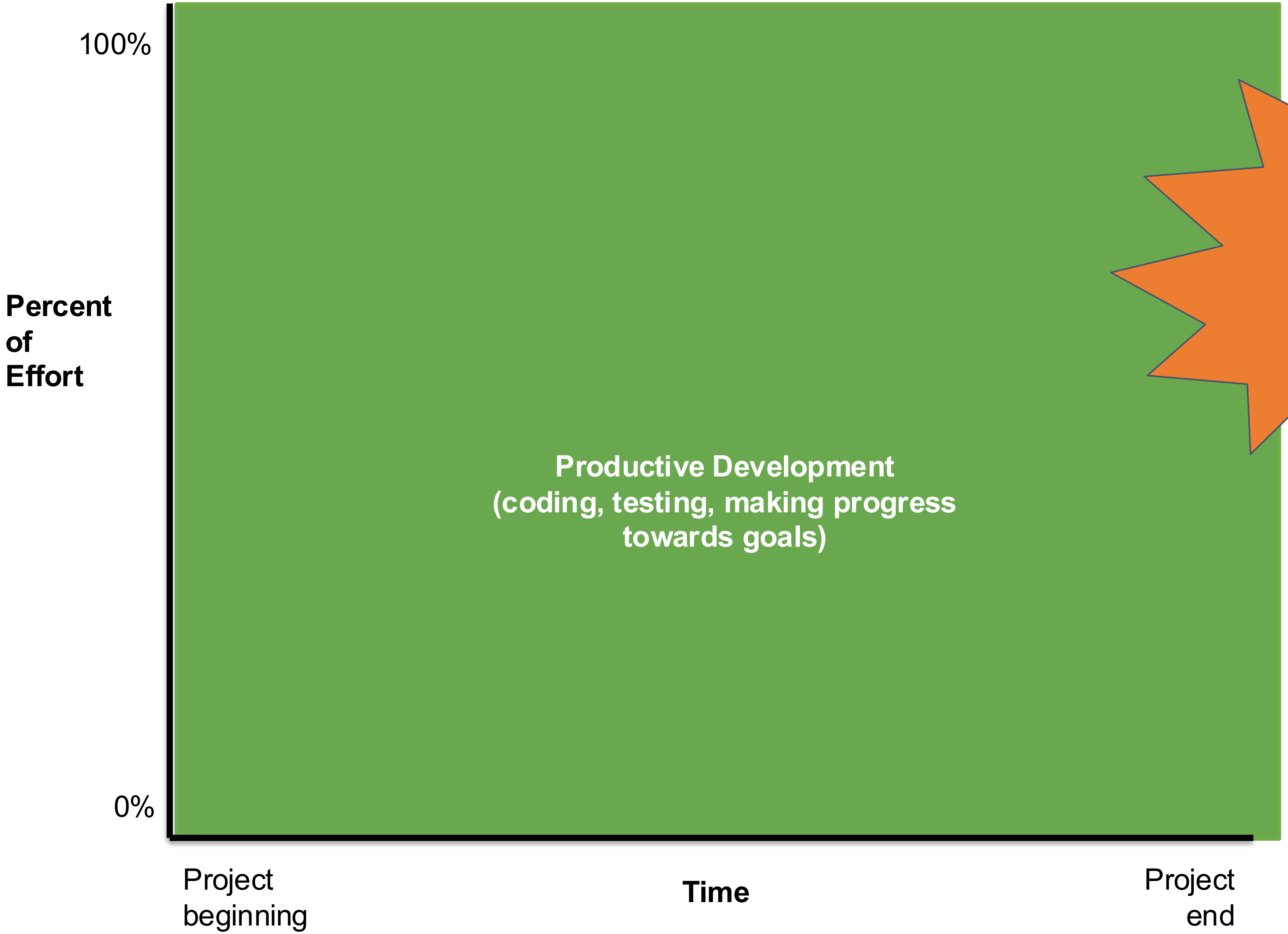
Debug to find causes of defects

Fix the defects

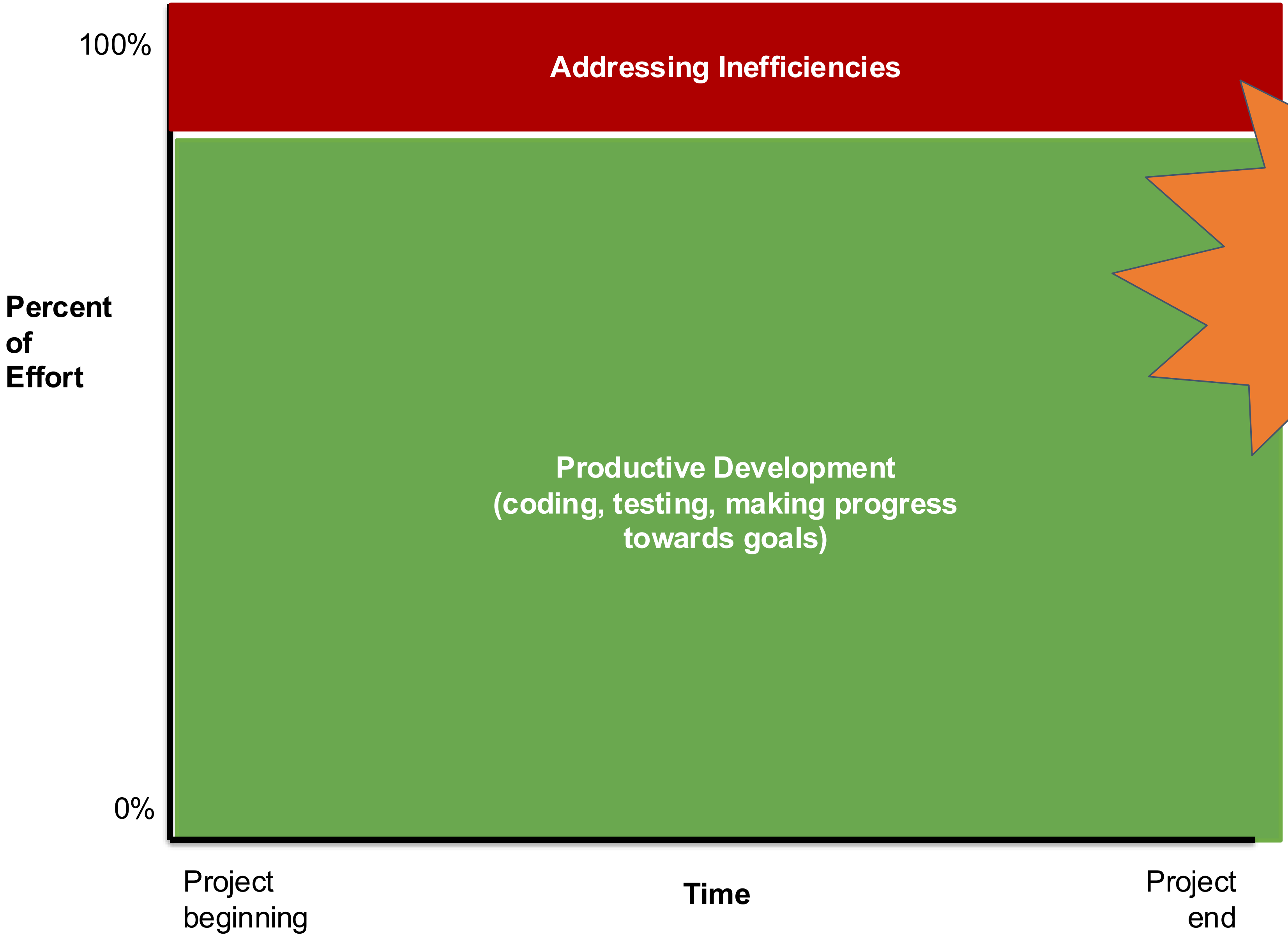
What does a software engineer's day look like?

- How many hours do they spend in meetings, coding, testing, debugging, etc.?



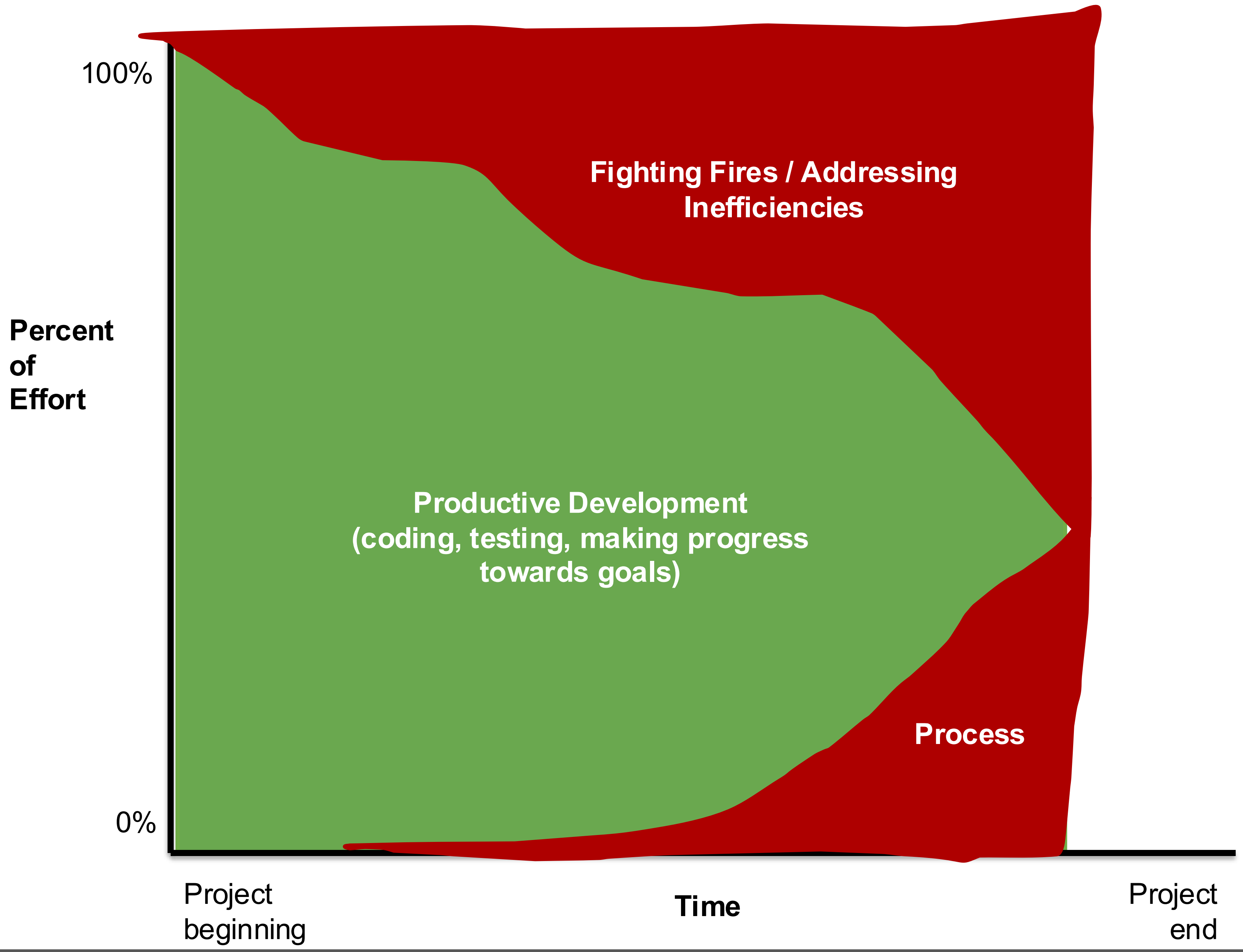

100%
Percent
of
Effort
0%
Productive Development
(coding, testing, making progress
towards goals)
Project
beginning
Time
Project
end

Idealized?


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

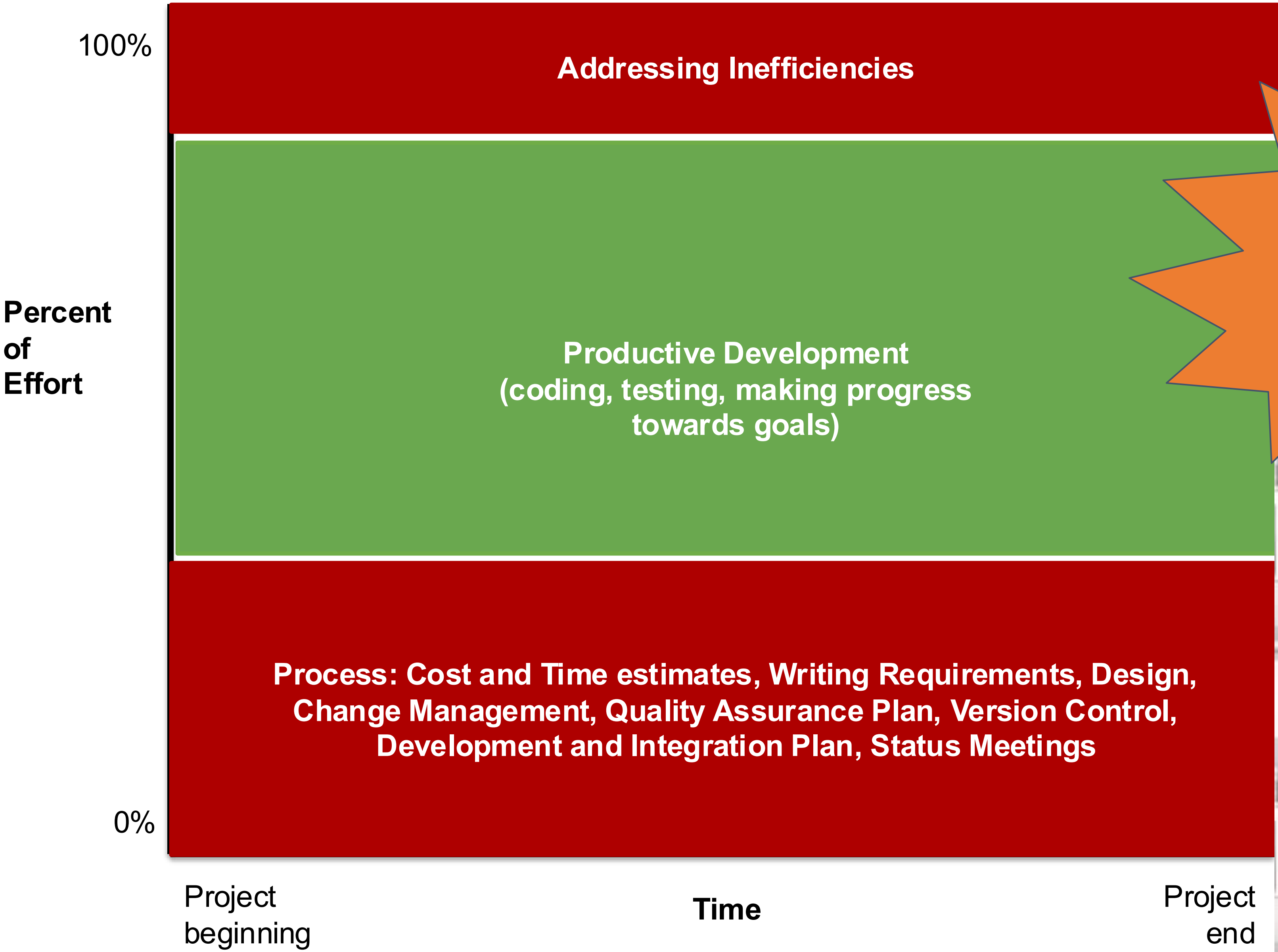
Idealized?

SO, TELL ME AGAIN
HOW DID P1 GO?


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

Let’s improve the reliability of this process

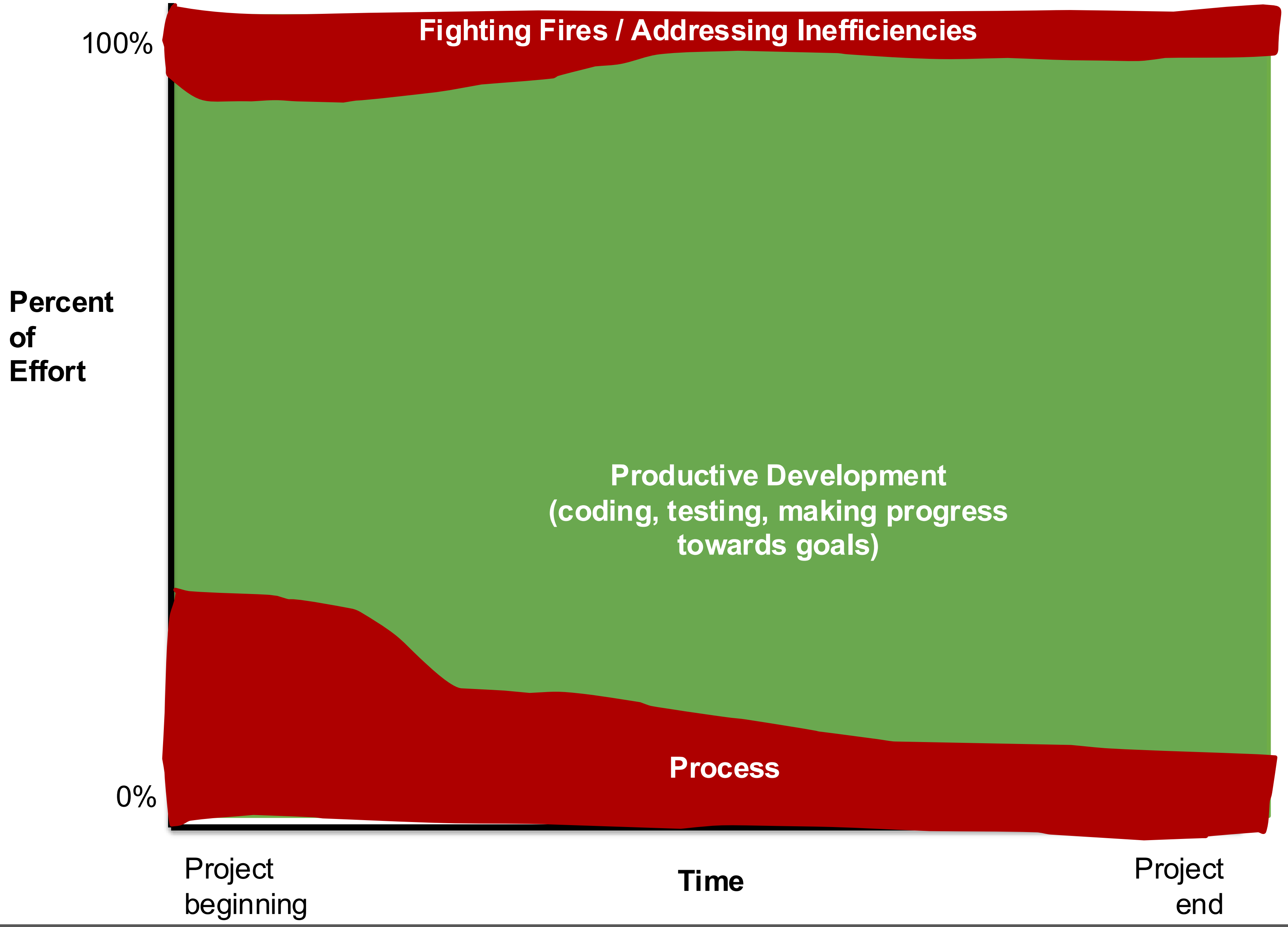
- Writing down all requirements
  - Review requirements
  - Require approval for all changes to requirements
- Use version control for all changes
  - Code Reviews
- Track all work items
  - Break down development into smaller tasks
  - Write down and monitor all reported bugs
  - Hold regular, frequent status meetings
- Plan and conduct quality assurance
- Employ a DevOps framework to push code between developers and operations



100%
Percent of Effort
0%
Addressing Inefficiencies
Productive Development (coding, testing, making progress towards goals)
Process: Cost and Time estimates, Writing Requirements, Design, Change Management, Quality Assurance Plan, Version Control, Development and Integration Plan, Status Meetings
Project beginning
Time
Project end

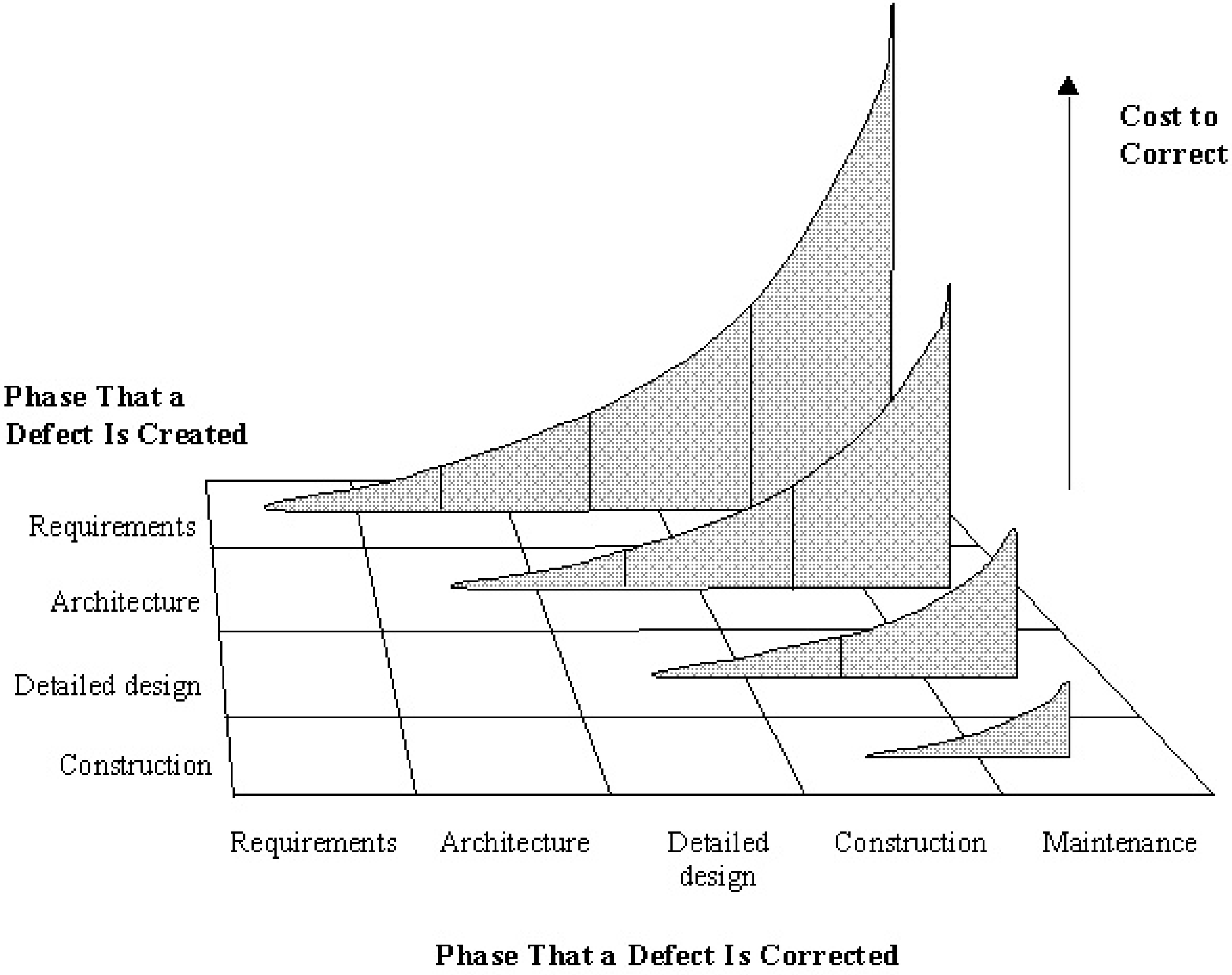
Negative View of Process

ACCEPT JOB AS SENIOR ENGINEER
8 HOURS OF MEETINGS A DAY
Posted in r/ProgrammerHumor
reddit



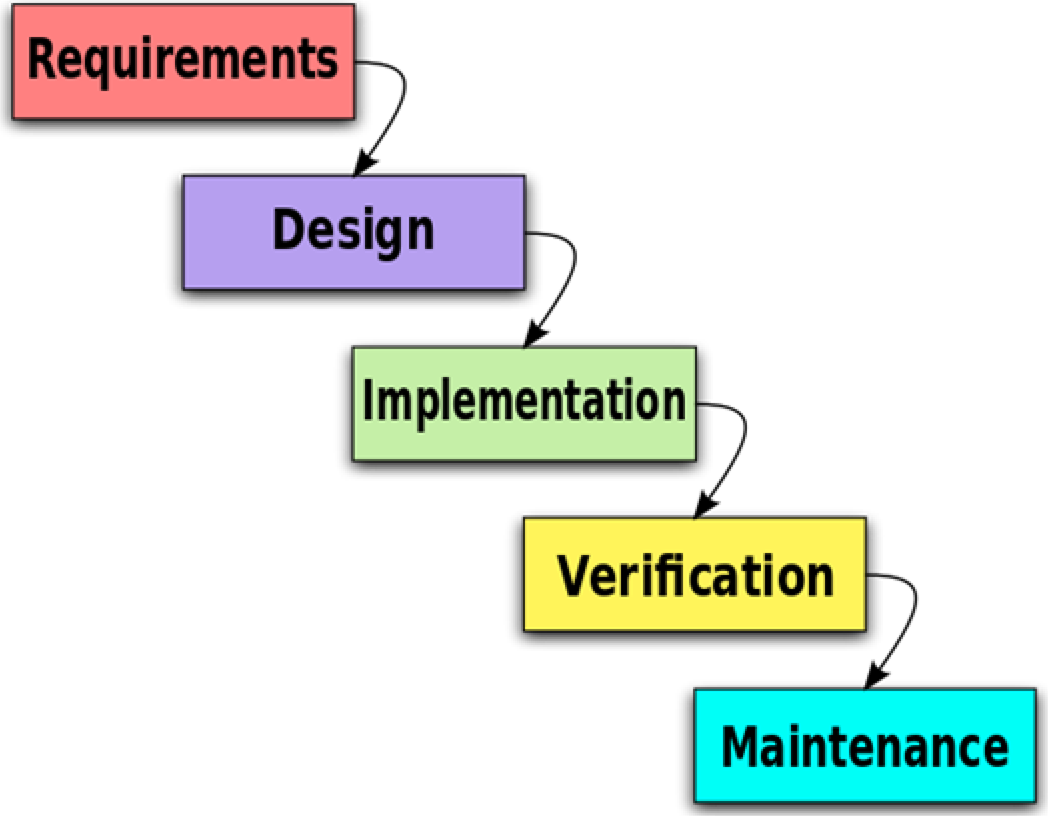
Hypothesis: Process increases flexibility and efficiency

Ideal Curve: Upfront investment for later greater returns



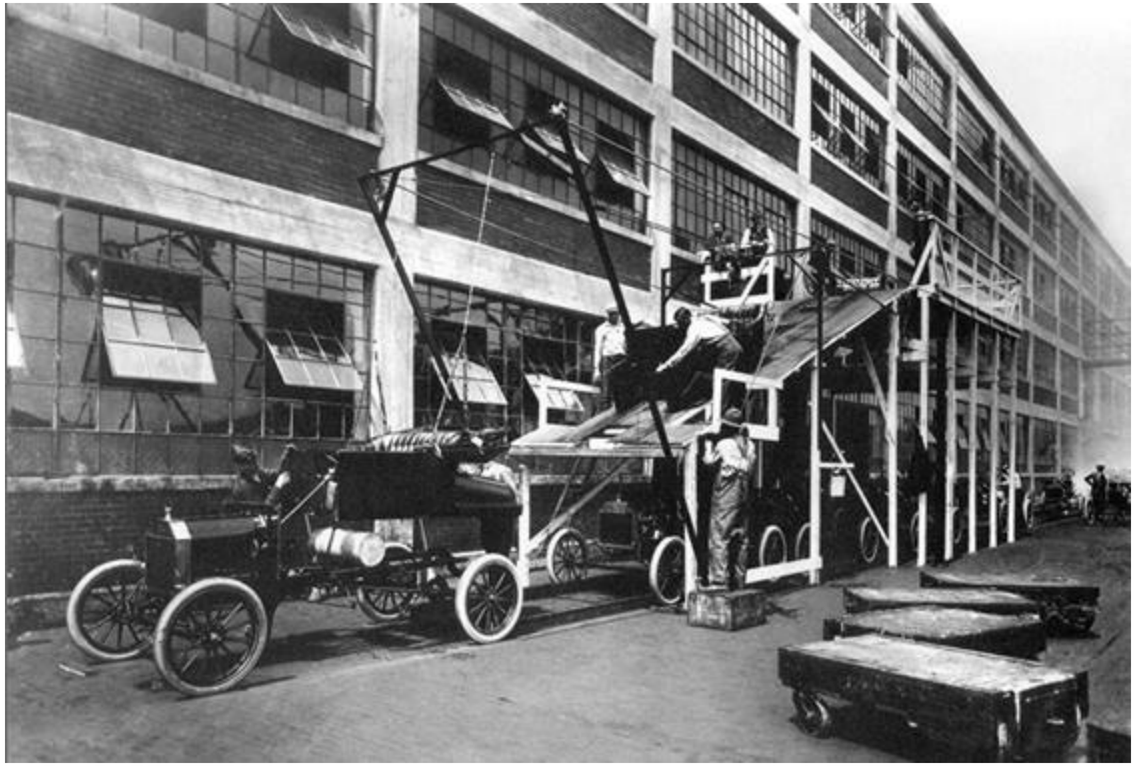
Copyright 1998 Steven C. McConnell. Reprinted with permission from *Software Project Survival Guide* (Microsoft Press, 1998).

Waterfall model was the original software process

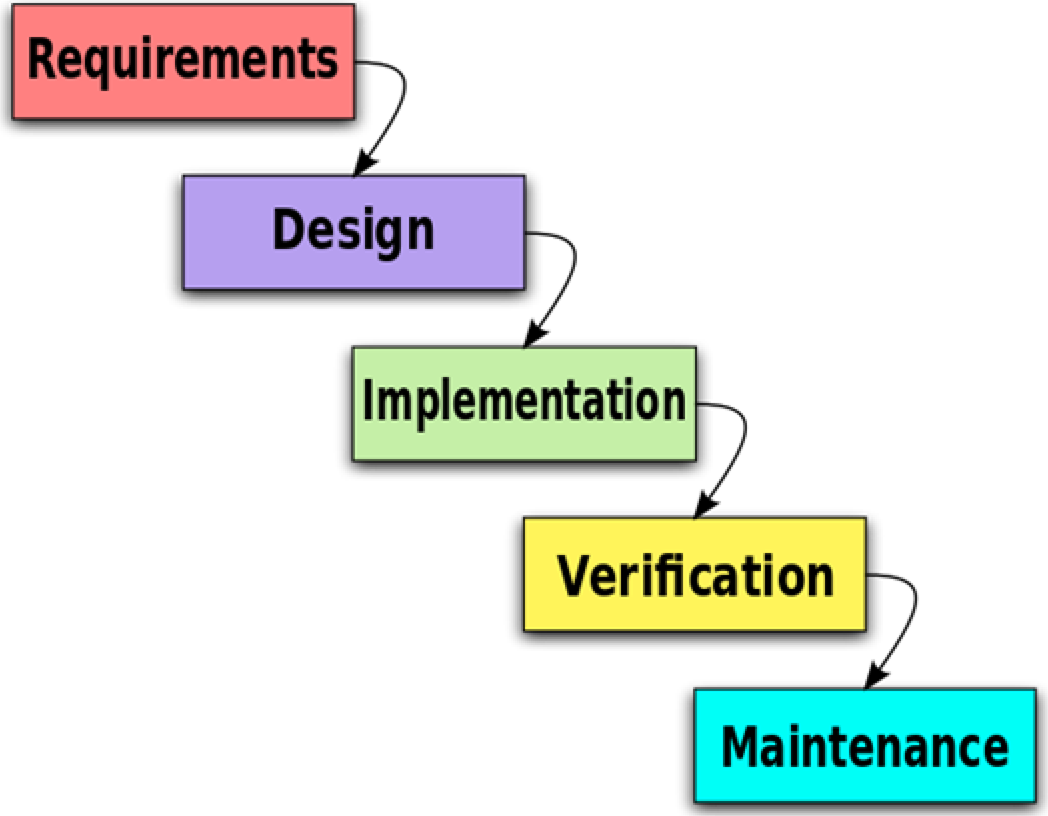


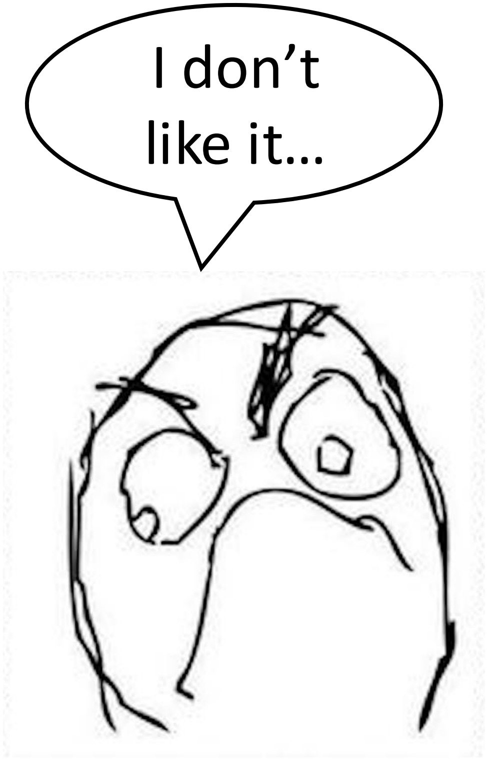
Waterfall diagram CC-BY 3.0 **Paulsmith99 at en.wikipedia**

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



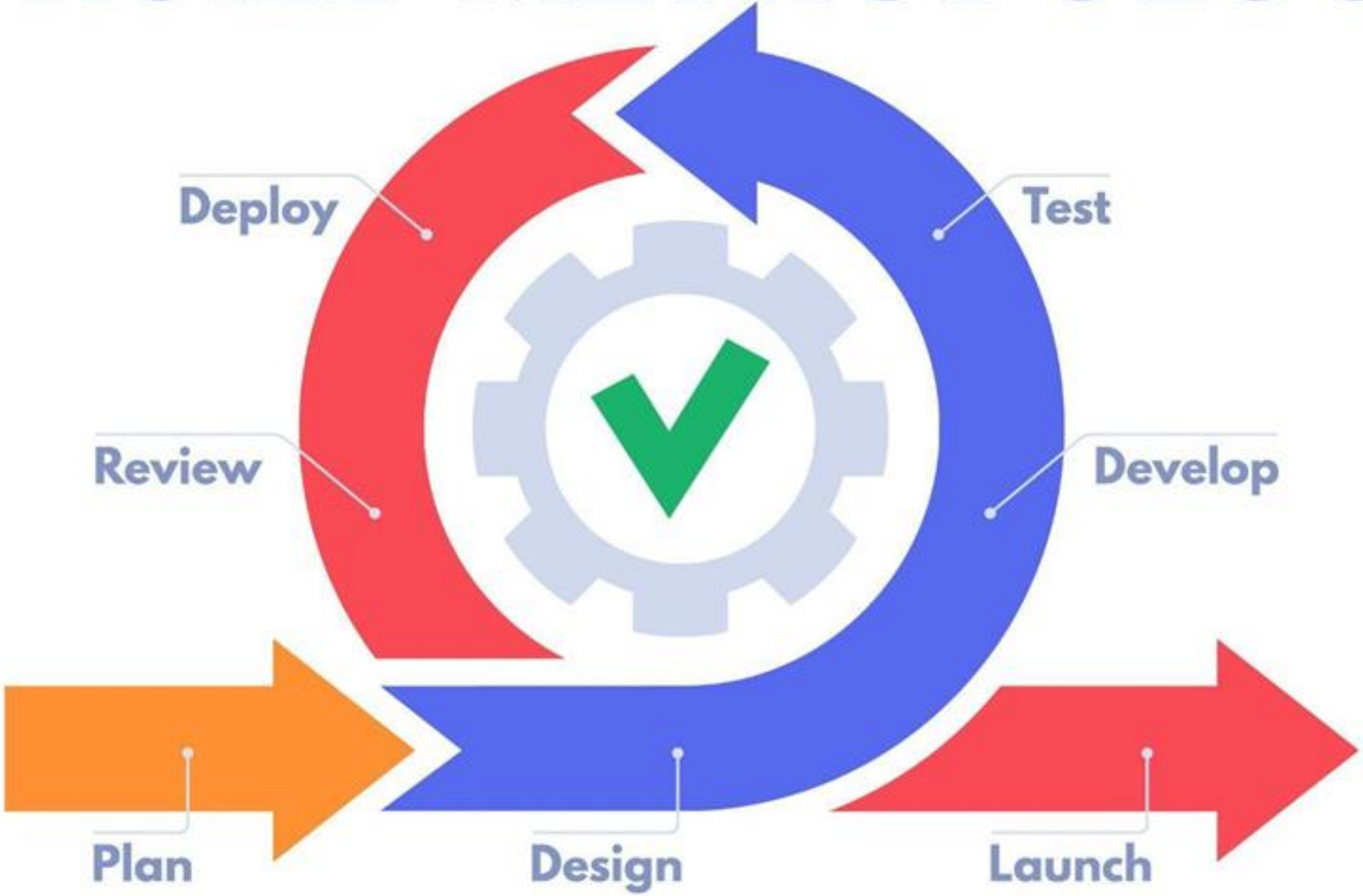
What could go wrong?





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

AGILE METHODOLOGY



Agile manifesto

- Twelve high-level principles
- e.g., *“Working software is the primary measure of progress”*

Scrum

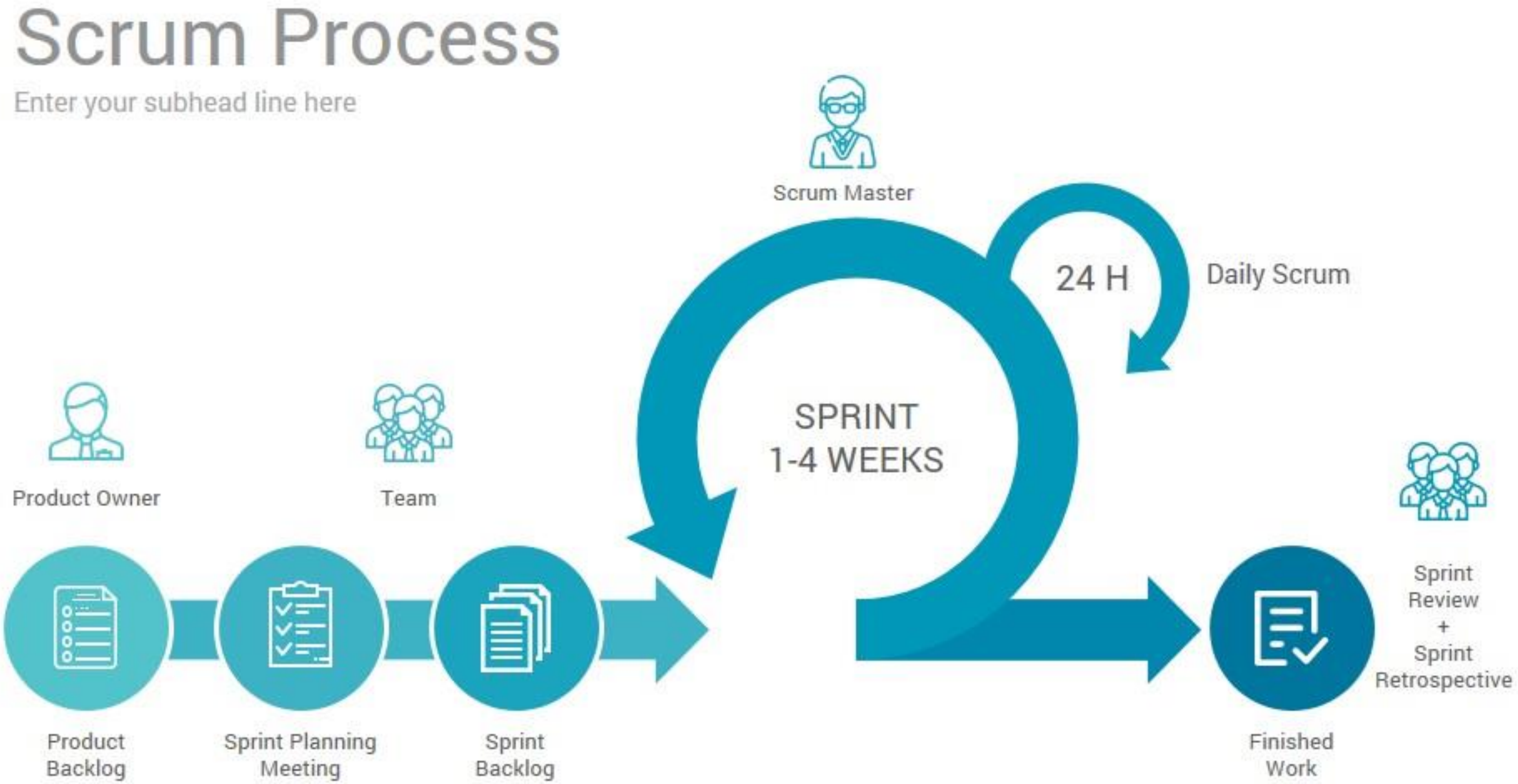
(Only a brief intro)

Agile Software Development with Scrum

red
yellow
green
blue
red
blue
yellow
green
blue

Ken Schwaber Mike Beedle

Elements of Scrum



Backlogs

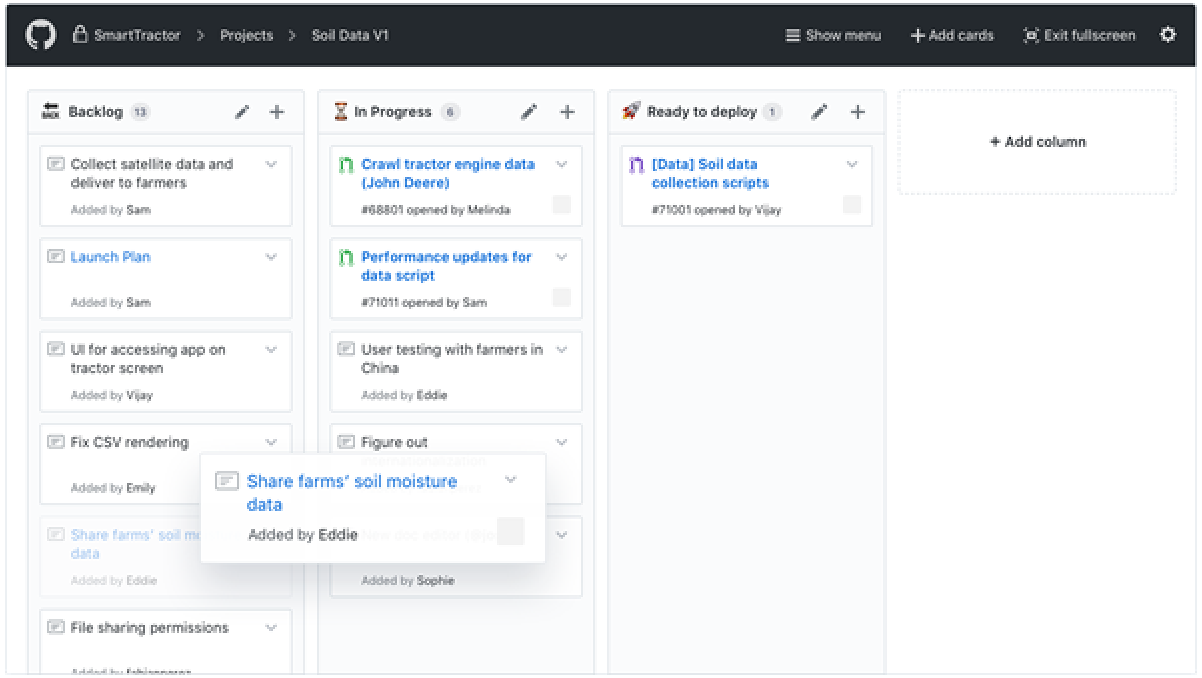
The **product backlog** is all the features for the product

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

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
- 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

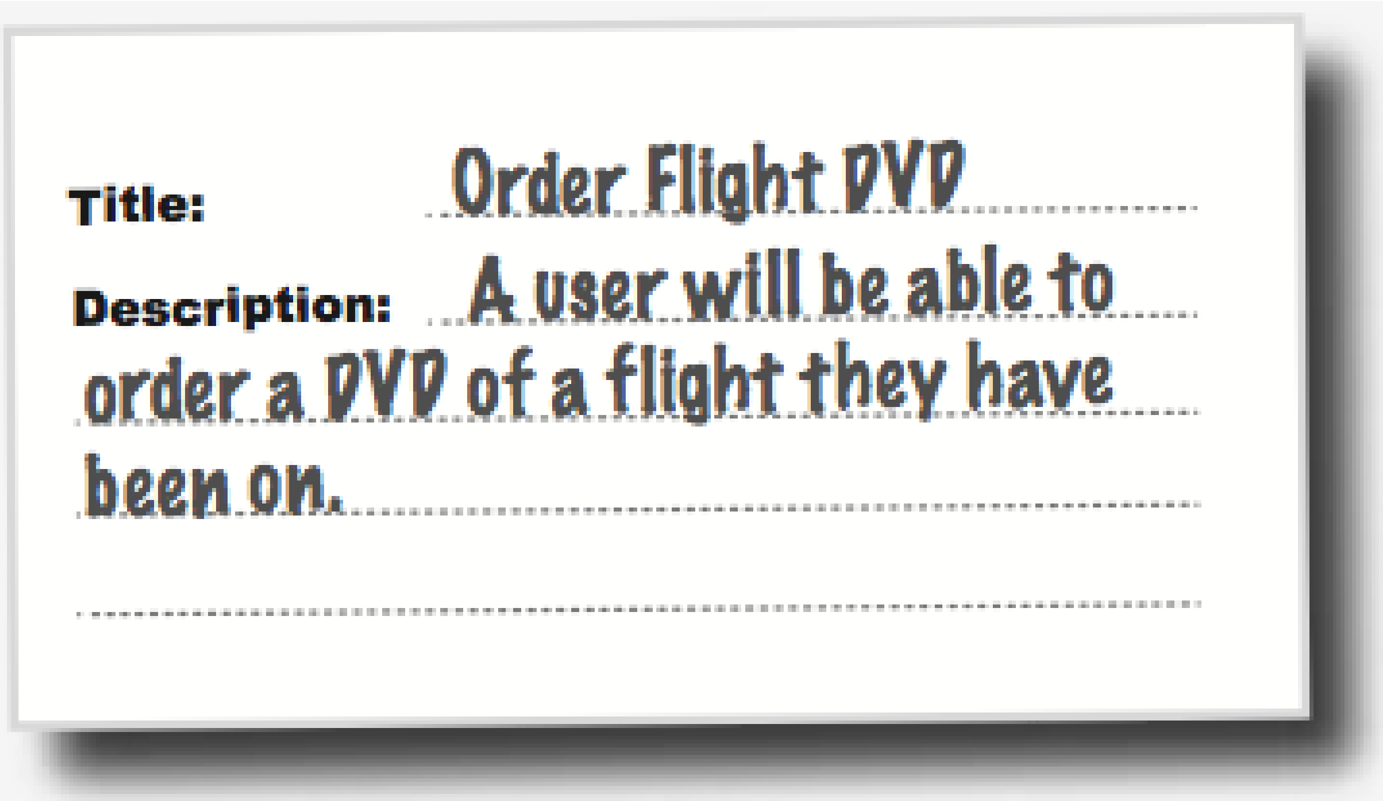
- Sprint Planning Meeting
    - Entire Team decides together what to tackle for that sprint
- Daily Scrum Meeting
    - Quick Meeting to touch base on :
        - What have I done? What am I doing next? What am I stuck on/need help?
- Sprint Retrospective
    - Review sprint process
- Sprint Review Meeting
    - Review Product

Standups

User stories

- Plan using units of customer-visible functionality

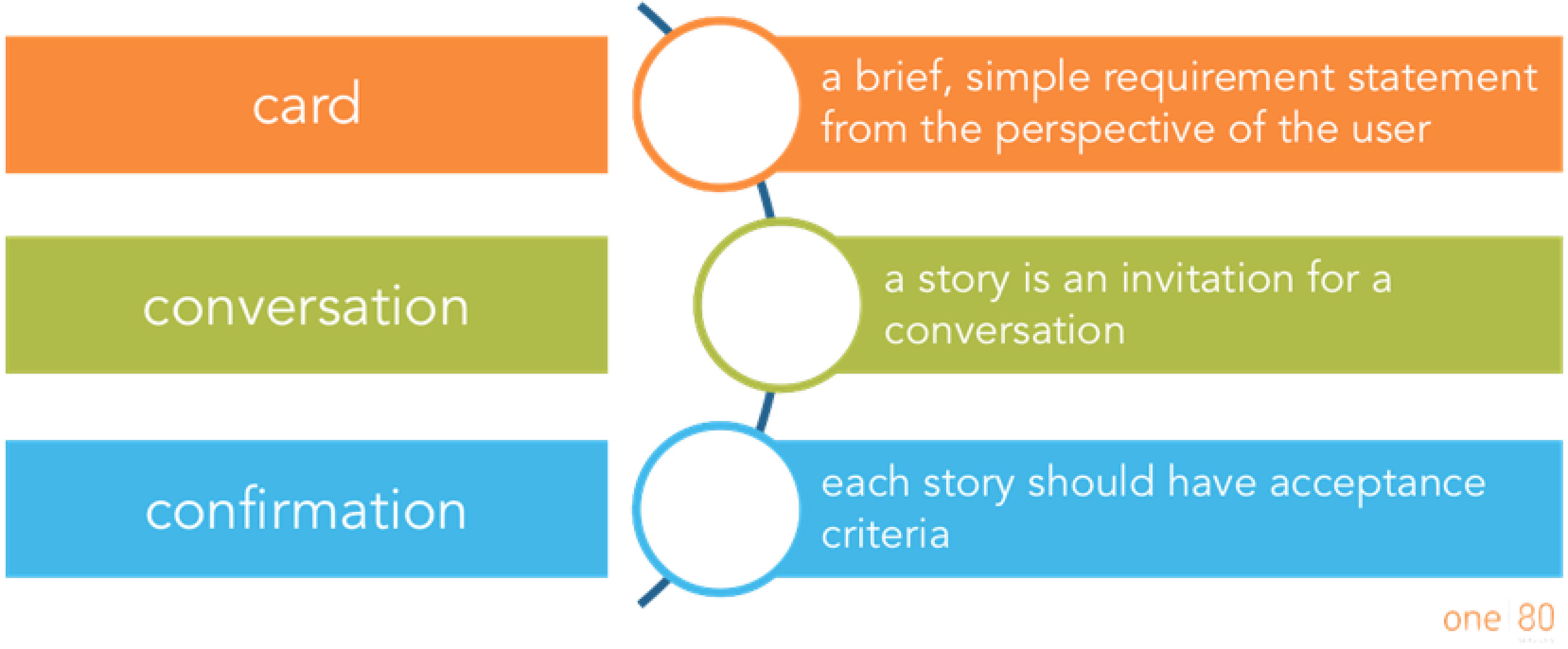
Example



Title: Order Flight DVD

Description: A user will be able to order a DVD of a flight they have been on.

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]"

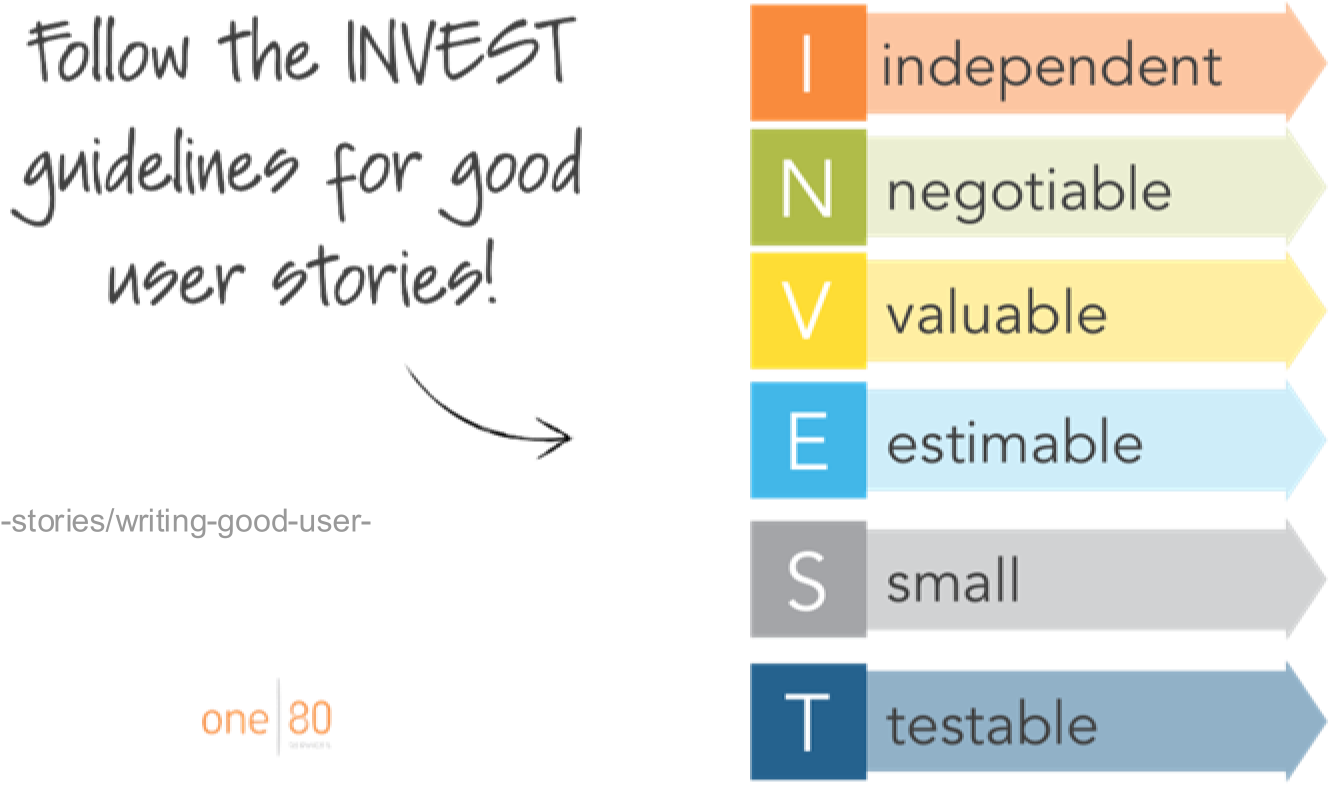
Conversation

- Developers, product managers, etc.
- Is it clear to everyone?
- What must a developer do to implement this user story?

Confirmation

- How can we tell that the user story has been achieved?
- It’s easy to tell when the developer finished the code.
- But, how do you tell that the customer is happy?

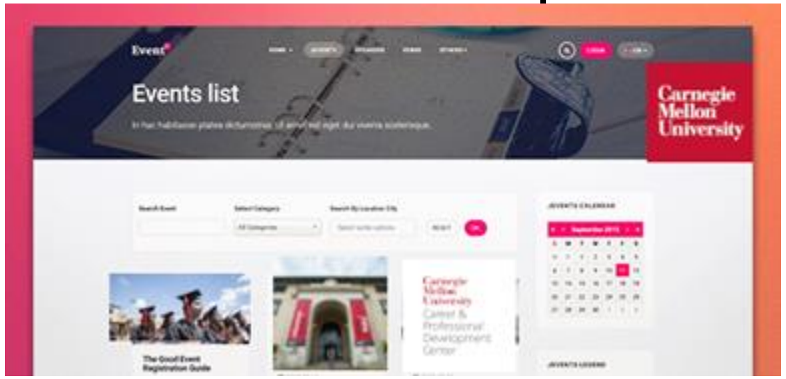
How to evaluate user story?



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

Example

The university is looking to enhance student and staff engagement by creating an online platform where all university-related events are easily accessible. The goal is to provide a user-friendly website that serves as a central hub for information on various activities, ranging from academic seminars to sports events and club meetings.



Independent

I independent

N negotiable

V valuable

E estimable

S small

T testable

- Schedule in any order.
- Not always possible

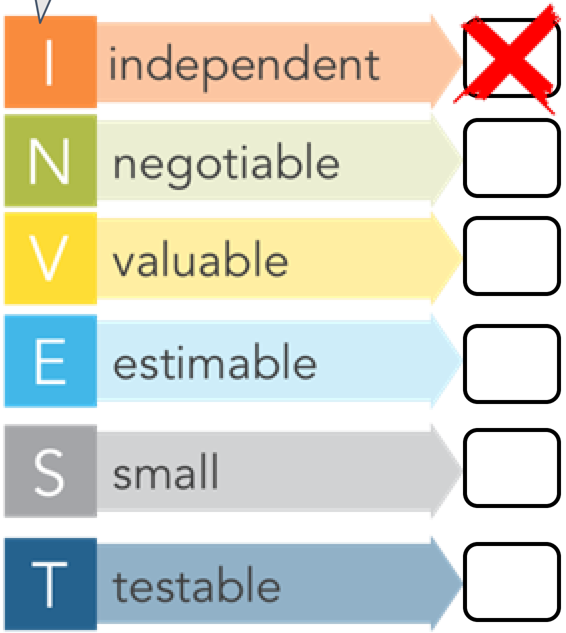
Counterexample

Assume that the homepage with an event calendar is already in place.

As a student, **I want to** receive notifications for events that are about to start, for those I have shown interest in, **so I** don't miss them.

Acceptance Criteria:

- An option is provided to 'Set a Reminder' for each event.
- Notifications are sent to users who have opted for reminders, shortly before the event starts.



I independent ☒

N negotiable □

V valuable □

E estimable □

S small □

T testable □

Negotiable

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

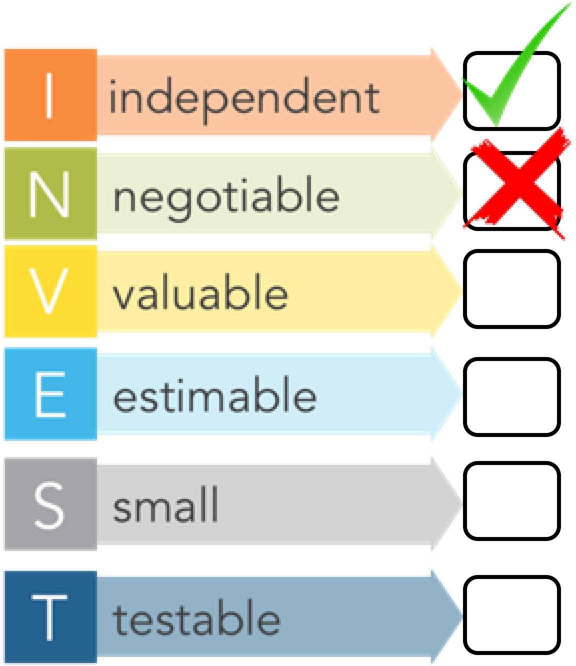
- Details to be negotiated during development
- Good Story captures the essence, not the details

Counterexample

As a student, **I want to** view the upcoming events at the university, **so I** can decide which ones to attend.

Acceptance Criteria:

- Add an interactive grid layout of upcoming events at the top of the homepage.
- Each event card in the grid is visible for a 2 seconds before automatically rotating to display the next set of events.
- Each card in the grid includes the event's name, type (e.g., seminar, sports game), duration, a brief description, and scheduled times.
- This grid of events is displayed under a prominent H1 heading that reads "Discover What's Happening on Campus!"



I independent ☑

N negotiable ☒

V valuable □

E estimable □

S small □

T testable □

Valuable

I independent

N negotiable

V valuable

E estimable

S small

T testable

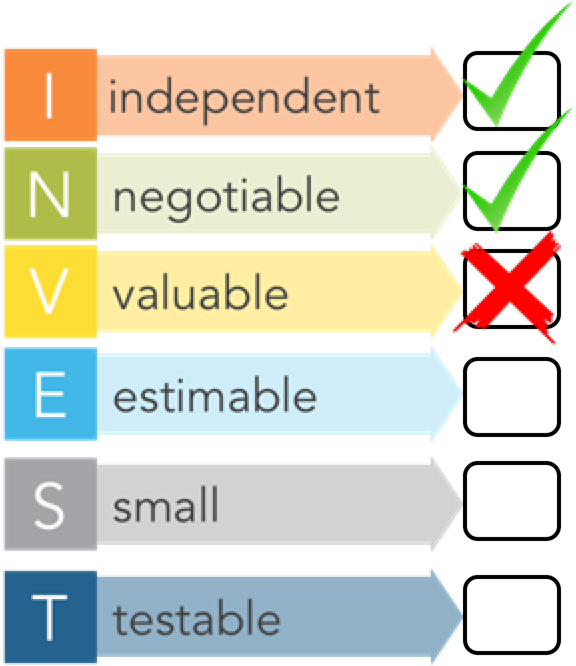
- This story needs to have value to someone (hopefully the customer)
- Easy to forget *why* you are doing what you are doing

Counterexample

As the Events Coordinator, **I want** a database to store details of students and staff interested in university events.

Acceptance Criteria:

- A database is constructed to manage user information.
- The database stores details such as name, email, phone number, favorite event types, date of birth, and history of event attendance or registrations.



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

Estimable

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

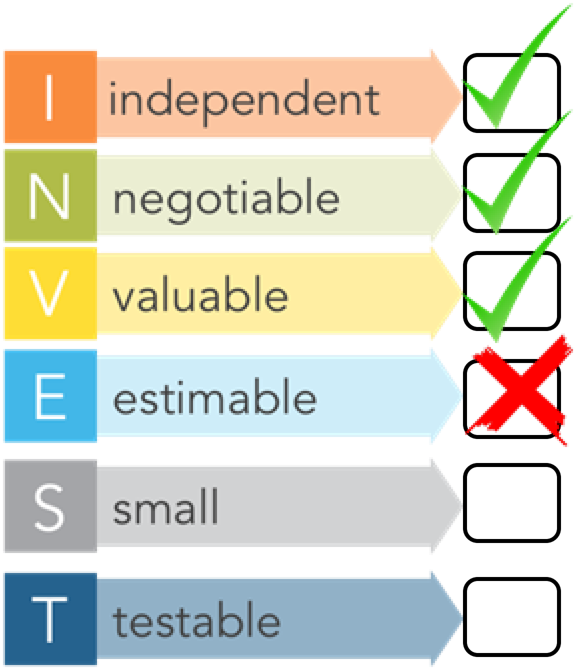
- Helps keep the size small
- It should provide enough details to estimate the amount of effort needed
- More on estimates later…

Counterexample

As an undergraduate student, **I want to** be able to filter university events, **so I** can choose the ones that align with my interests.

Acceptance Criteria:

- Filters are added to the event listings on the website.



I independent ☑

N negotiable ☑

V valuable ☑

E estimable ☒

S small □

T testable □

Small

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

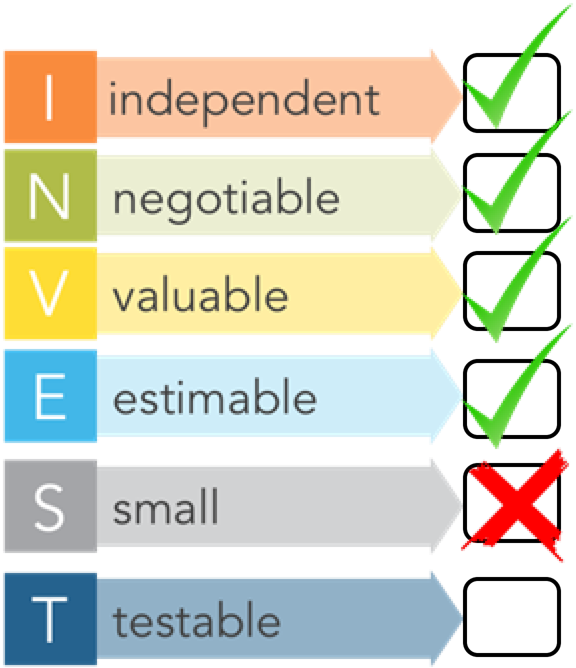
- Fit on 3x5 card
- At most two person-weeks of work (one sprint)
- Too big == unable to estimate

Counterexample

As a student, **I want to** easily find information about upcoming events, **so I** can participate in activities that interest me.

Acceptance criteria:

- A homepage is created displaying the university's name, motto, location, email, and contact information.
- The homepage features a calendar of upcoming university events.
- The event calendar includes details such as the event title, type (e.g., seminar, sports game, club meeting), a brief description, location, date, and time.
- Users can filter the event list by event type, date, and hosting department or club.
- The admin can update the event calendar as new events are planned or existing events are modified.



I independent ☑

N negotiable ☑

V valuable ☑

E estimable ☑

S small ☒

T testable □

Testable

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

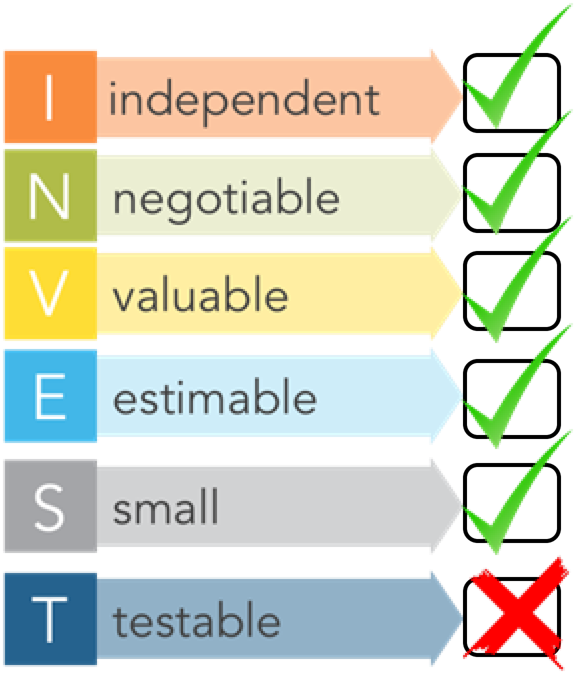
- Ensures understanding of task
- We know when we can mark task “Done”
- Unable to test == do not understand

Counterexample

As a student, **I want to** easily view promotional videos or trailers of university events, **so I** can decide which events to attend.

Acceptance Criteria:

- Promotional videos can be embedded on each event detail page.
- Videos are of high quality.
- The embedded video is well-integrated into the page design.
- The video size is large enough to ensure clarity.
- The video controls are user-friendly.



I independent ✓

N negotiable ✓

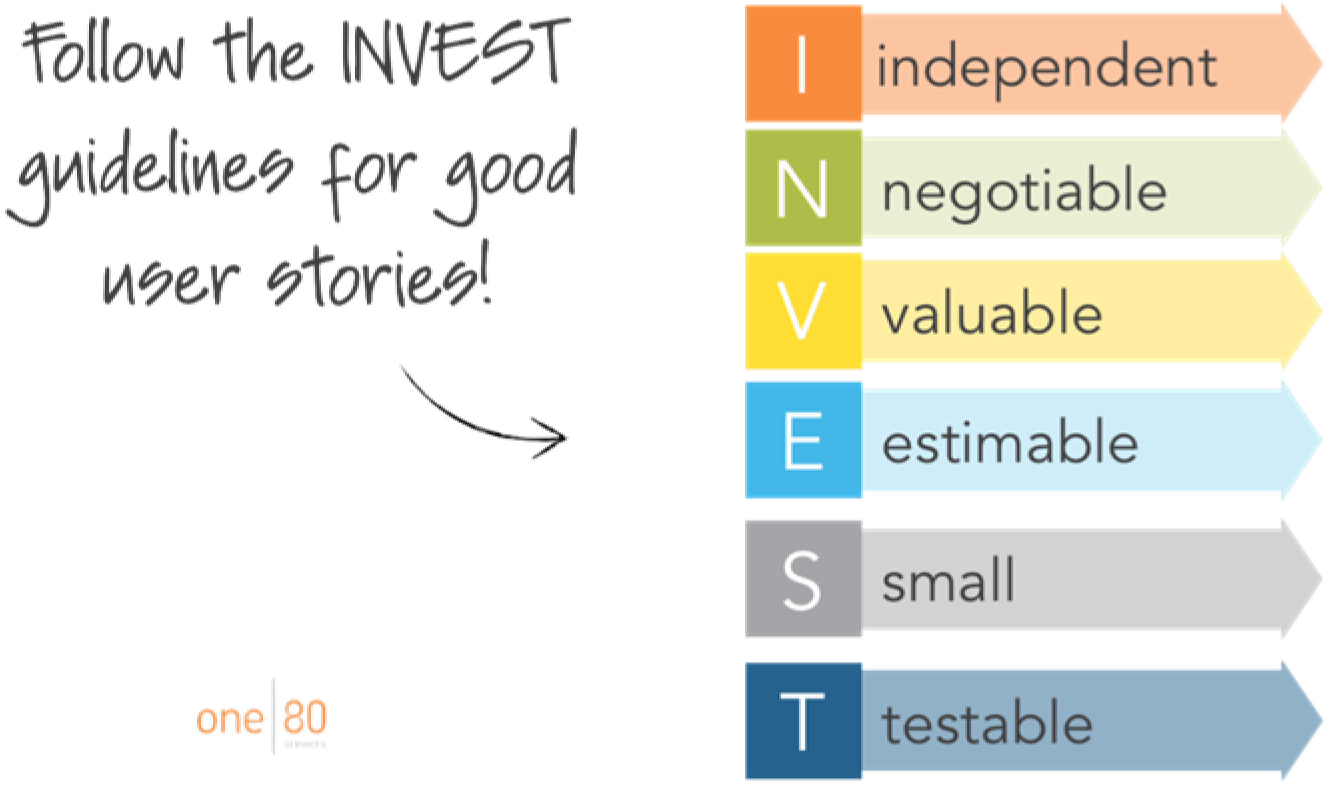
V valuable ✓

E estimable ✓

S small ✓

T testable ✗

Activity: Evaluate using INVEST

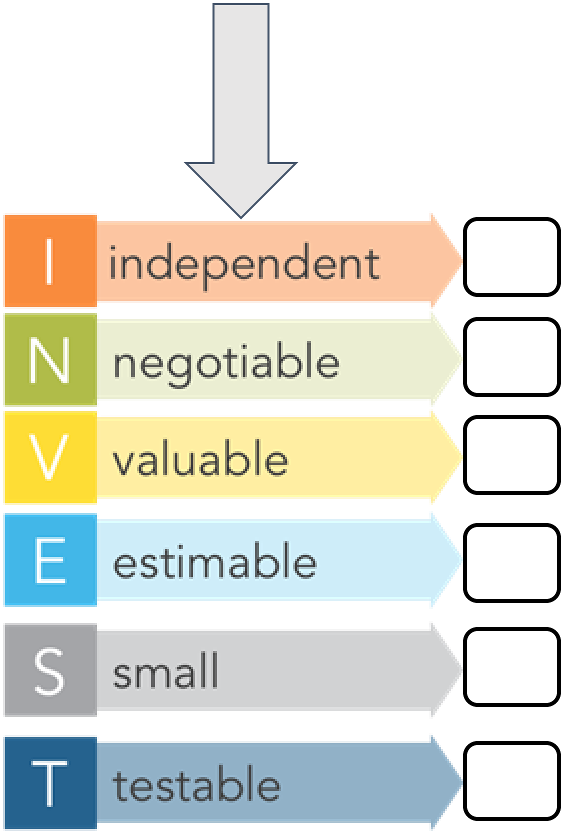


Follow the INVEST guidelines for good user stories!

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

User Story #1

Select the most serious flaw



I independent □

N negotiable □

V valuable □

E estimable □

S small □

T testable □

As the Events Coordinator, **I want** the website to seamlessly integrate with various academic calendars and departmental schedules, **so that** event information is always synchronized and accurate.

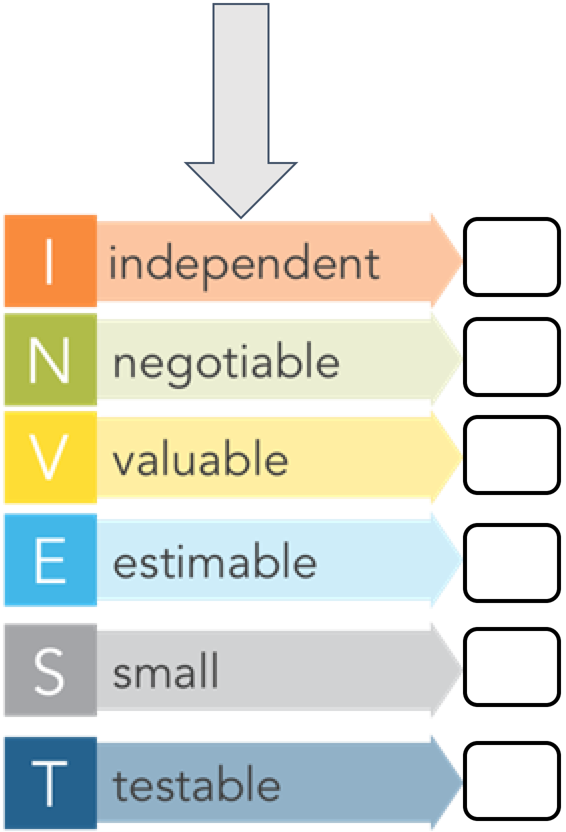
Acceptance Criteria:

- The website integrates with different academic and departmental calendars.
- Event information on the website reflects real-time updates from these calendars.

How can you fix it?

User Story #2

Select the most serious flaw



As a student, **I want** the website to have an intuitive navigation system **so that** I can find events effortlessly.

Acceptance Criteria:

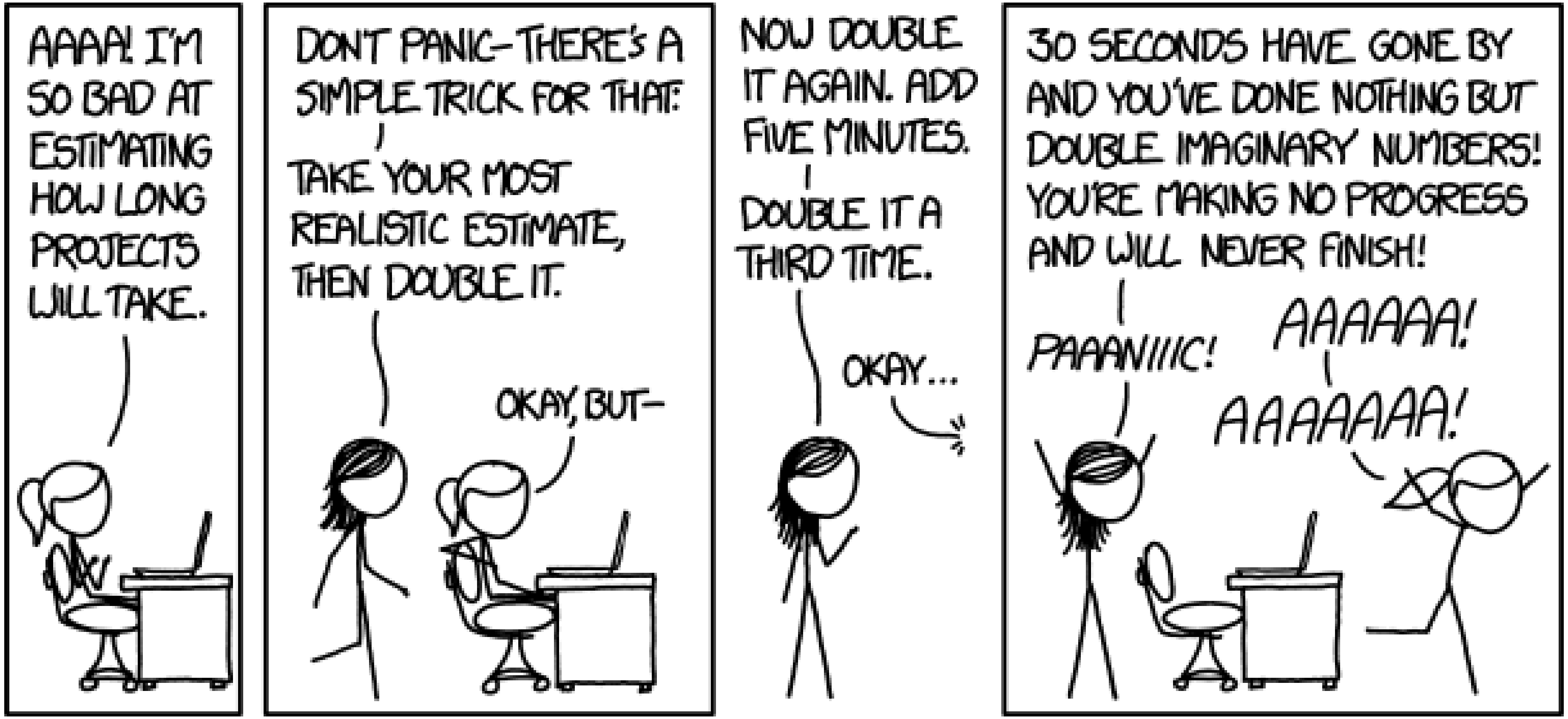
- The website's navigation is intuitive to users.
- Users can find events with minimal effort.
- The navigation system feels natural and easy to understand.

How can you fix it?

"Plans are nothing, planning is everything"

-Dwight D. Eisenhower

Time estimation


AAAA! I'M SO BAD AT ESTIMATING HOW LONG PROJECTS WILL TAKE.
DON'T PANIC—THERE'S A SIMPLE TRICK FOR THAT:
TAKE YOUR MOST REALISTIC ESTIMATE, THEN DOUBLE IT.
OKAY, BUT—
NOW DOUBLE IT AGAIN. ADD FIVE MINUTES.
DOUBLE IT A THIRD TIME.
OKAY...
30 SECONDS HAVE GONE BY AND YOU'VE DONE NOTHING BUT DOUBLE IMAGINARY NUMBERS! YOU'RE MAKING NO PROGRESS AND WILL NEVER FINISH!
PAAANIIIC!
AAAAAA!
AAAAAAA!

https://xkcd.com/1658/

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, part 2

Review your estimates

- Still think they are accurate?
- Any changes?
- Get feedback from 2 other people

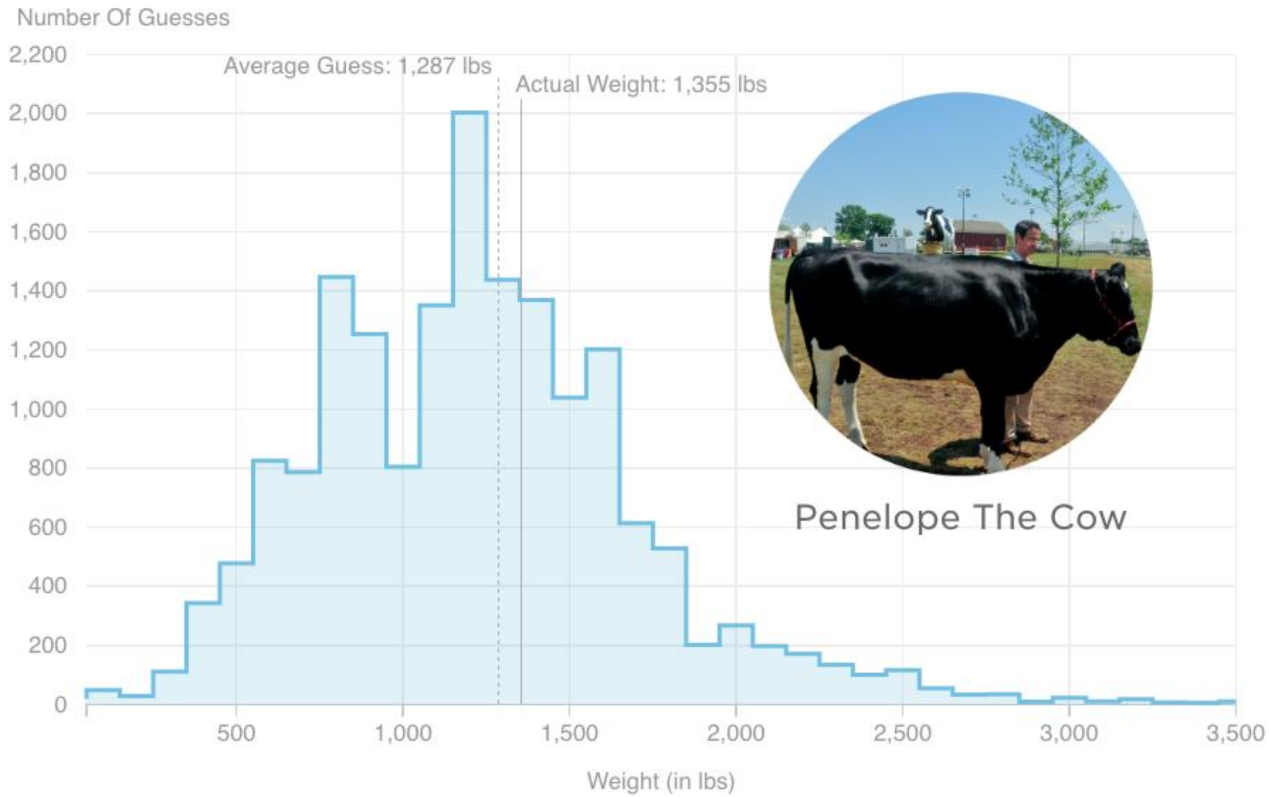
Improving Time Estimates

- Prevent conformity bias
- Do you have a comparable experience to base an estimate on?
- How much design do you need for each task?
- Break down the task into smaller tasks and estimate them.

Wisdom of the Crowd

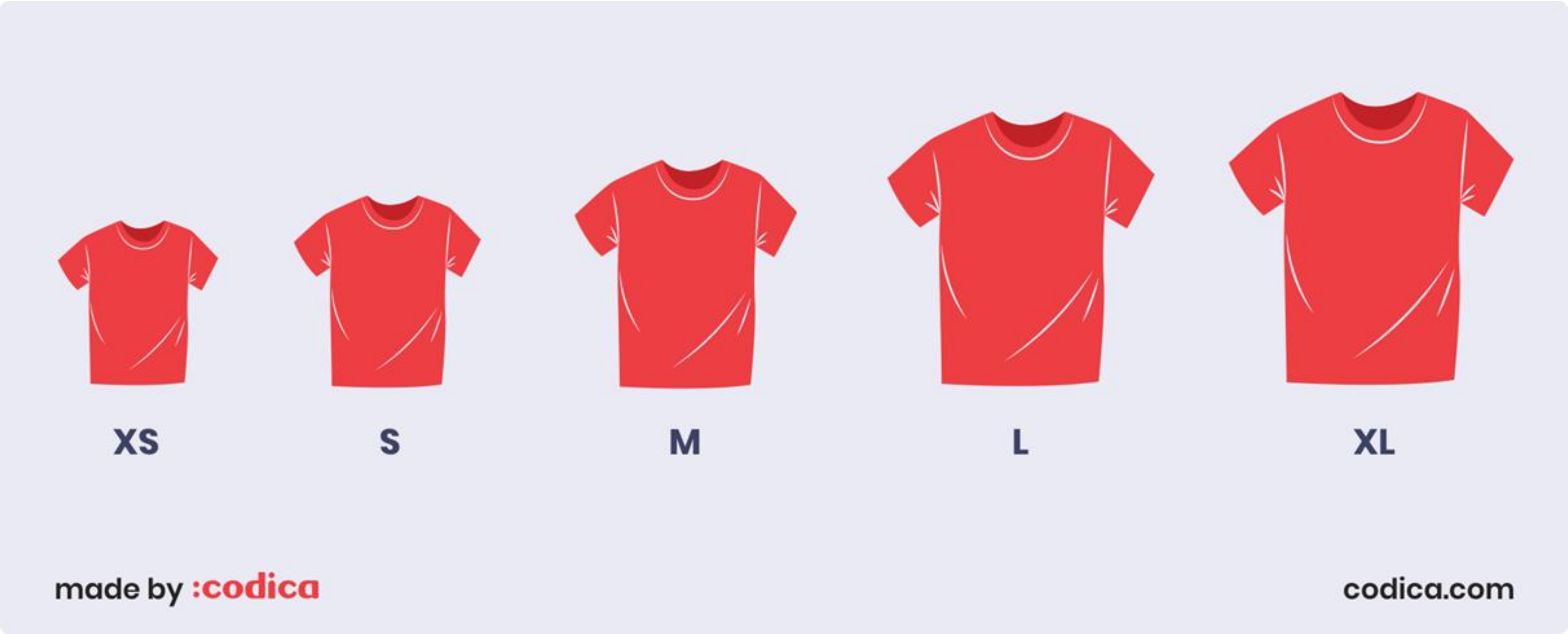
How Much Does This Cow Weigh?

(All People)



Source: The Internet.

Credit: Quoctrung Bui/NPR

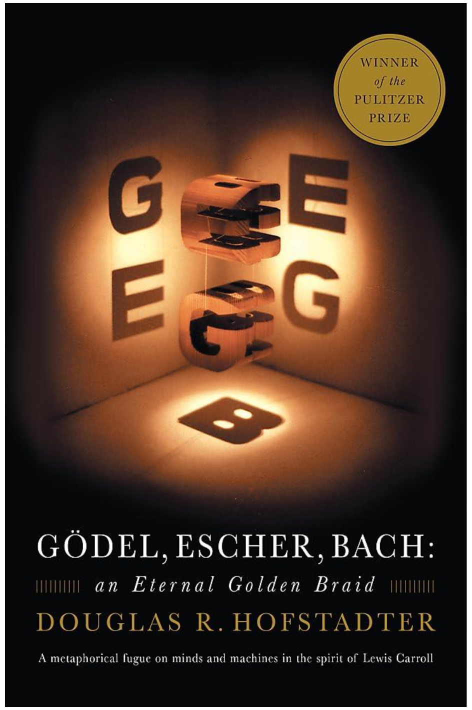

XS
S
M
L
XL
made by :codica
codica.com

1
2
3
5
8
13
21

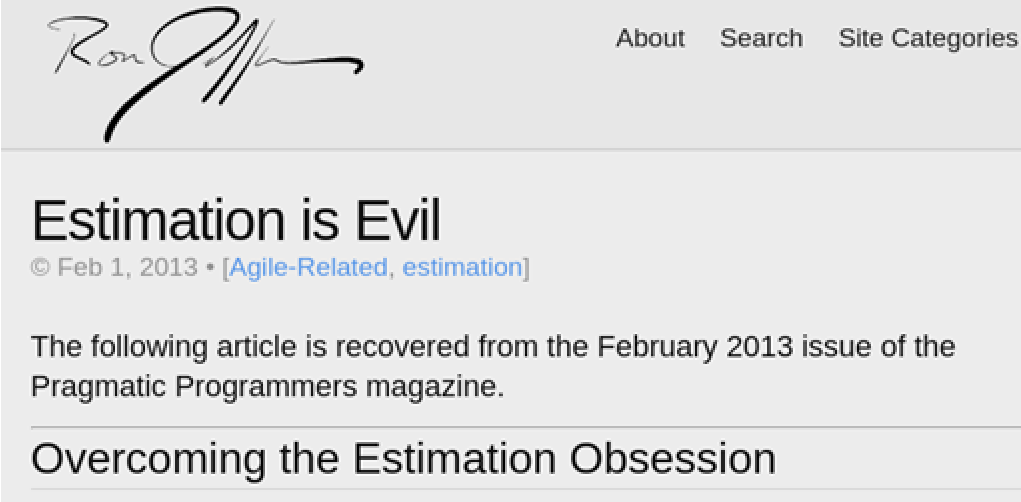
$\times\pi$

Hofstadter’s Law

“It always takes longer than you expect, even when you take into account Hofstadter’s Law”


WINNER of the PULITZER PRIZE
GÖDEL, ESCHER, BACH:
an Eternal Golden Braid
DOUGLAS R. HOFSTADTER
A metaphorical fugue on minds and machines in the spirit of Lewis Carroll

Is Estimation Evil?

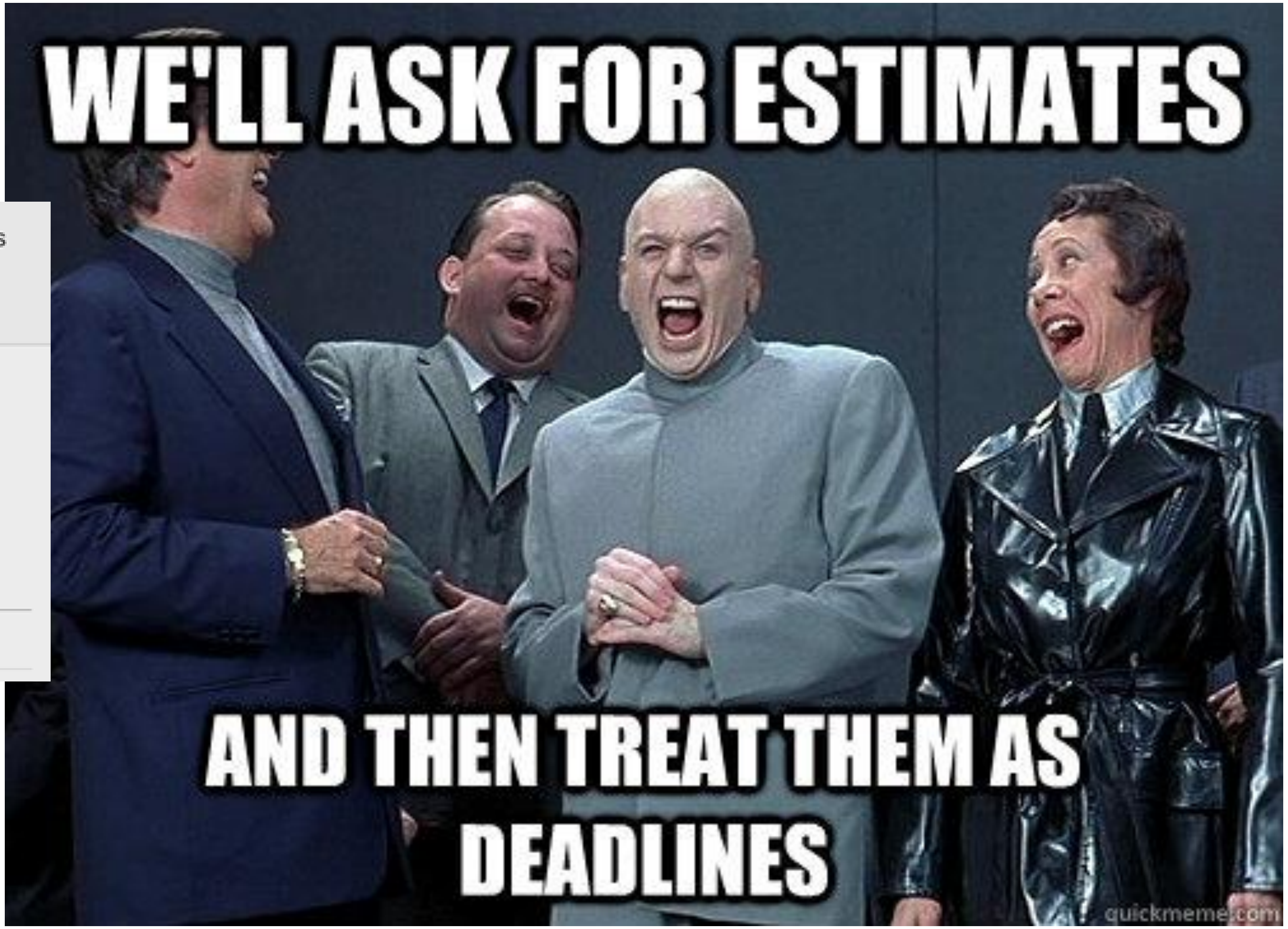


Estimation is Evil

© Feb 1, 2013 • [Agile-Related, estimation]

The following article is recovered from the February 2013 issue of the Pragmatic Programmers magazine.

Overcoming the Estimation Obsession



Ron Jeffries's essay Estimation is Evil

Milestones and deliverables make progress *observable*

Milestone: clear end point of a (sub)tasks

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

Deliverable: Result for customer

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

What you need to know

- Recognize the importance of having a software process
- Main ideas of Agile/Scrum
- Understand backlogs and user stories
- Understand the difficulty of estimating tasks and progress
- We use milestones for planning and progress measurement