Intro To Process
Milestones, Estimation, Planning

17-313 Spring 2024

Foundations of Software Engineering

https://cmu-313.github.io

Michael Hilton and Eduardo Feo Flushing

Administrivia

- Project 2(a) due on Thursday (Feb 1st) at midnight
- Meet with your teams!
- Extra credit: Go out with your teams socially.
  - Share a photo/screenshot of your team activity with your TA before Thursday night.

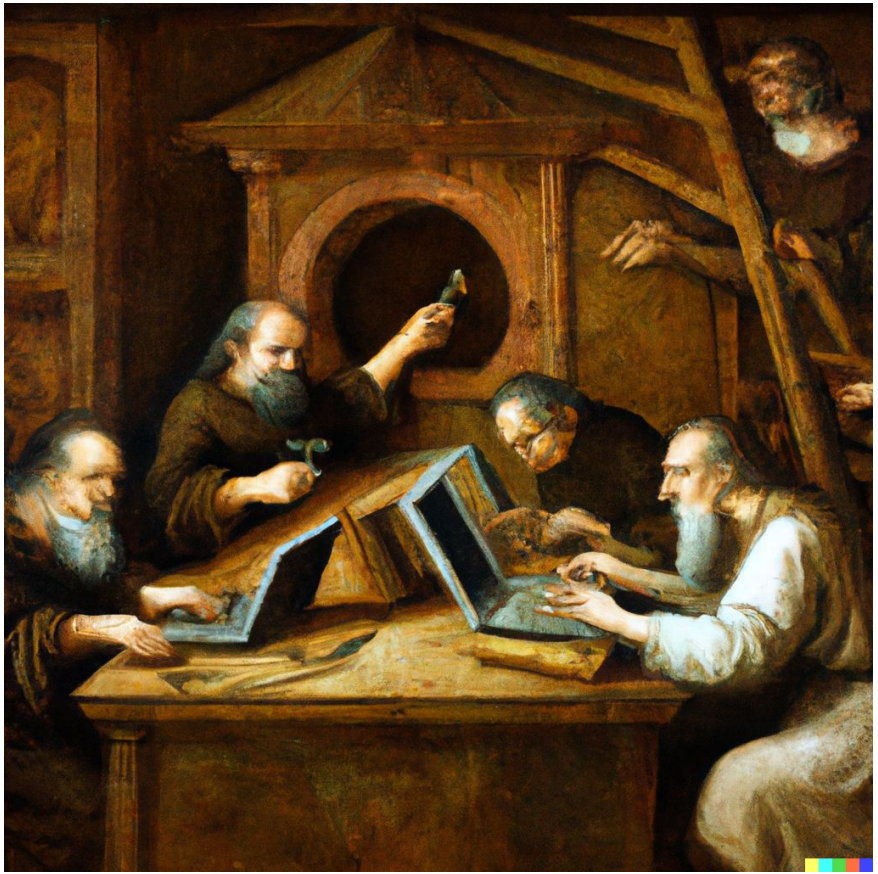
Smoking Section

- Last full row

Today's Learning Goals

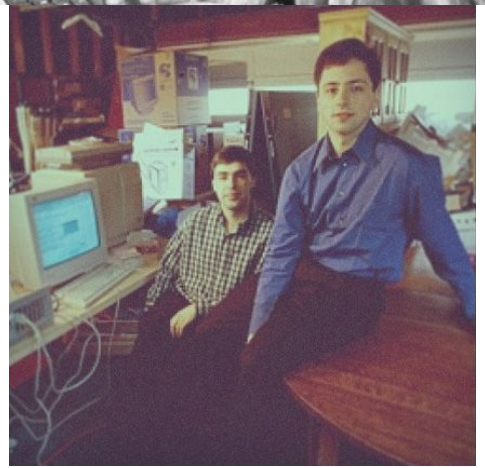
- Recognize the importance of process
- Identify why software development has project characteristics
- Understand the elements of Scrum
- Create and evaluate user stories
- Use milestones for planning and progress measurement
- Understand the difficulty of measuring progress

Software Development… before Software Engineering



by DALL-E


PayPal



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

Software Eng[illegible]rinciples, practices (technical and non-technical) for confidently building high-quality software.

Outline

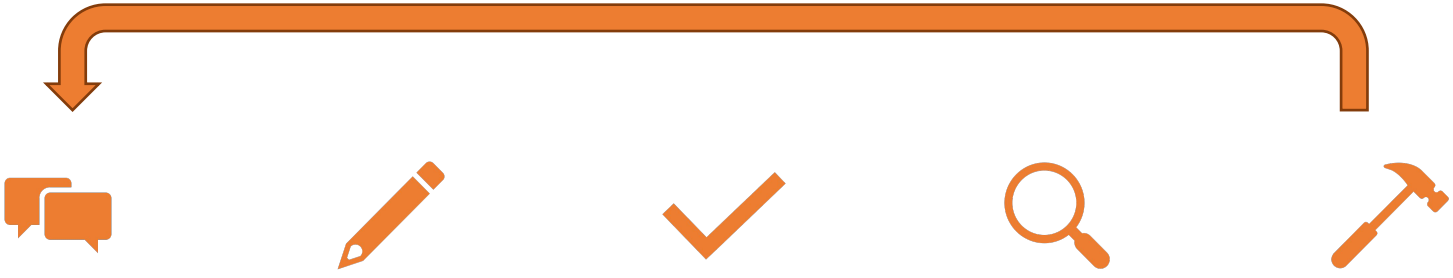
- Software Processes and why we need them
- Software Process Models
  - Agile and Scrum
- Planning: Task and progress estimation

Software Process

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

Sommerville, SE, ed. 8

How to develop software???



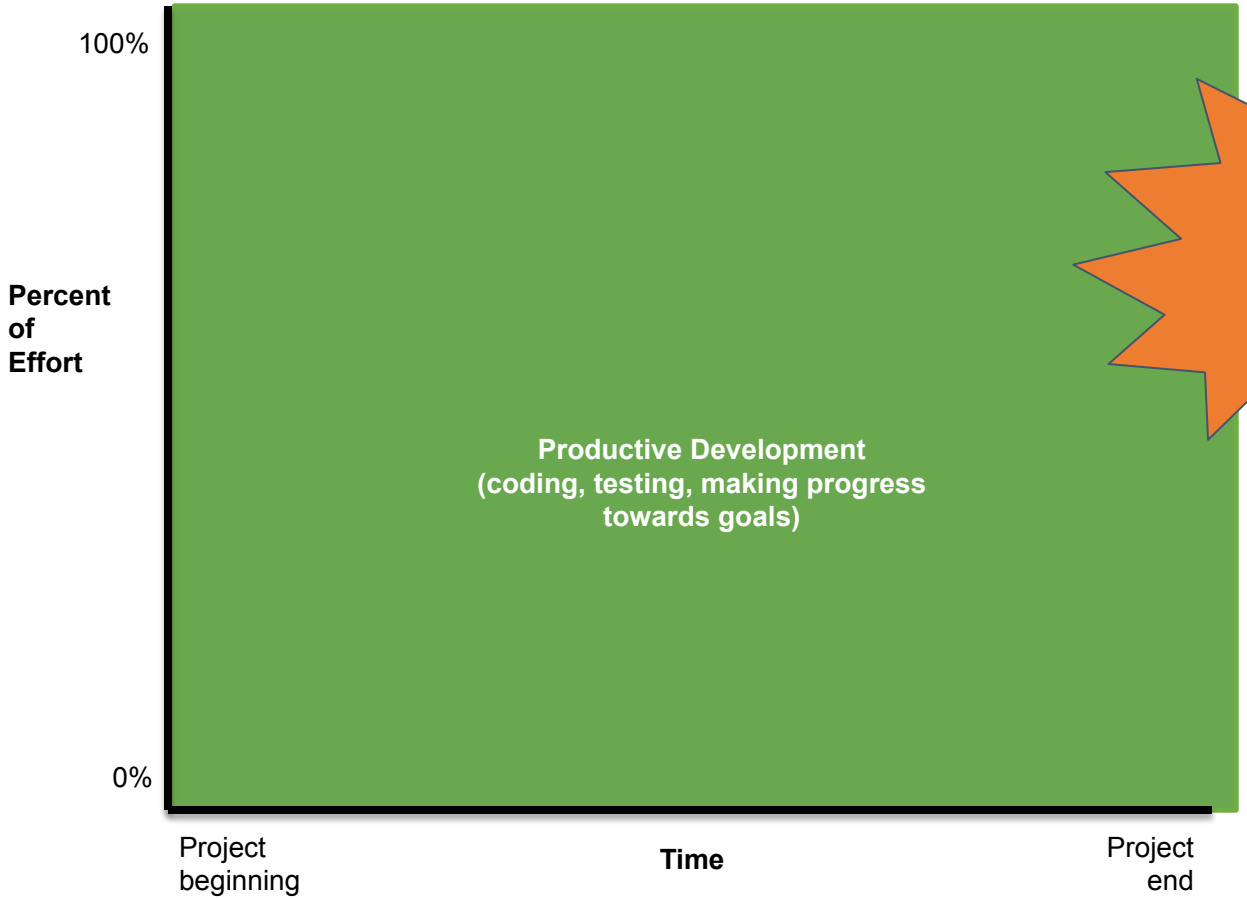
Discuss the software that needs to be written

Write some code

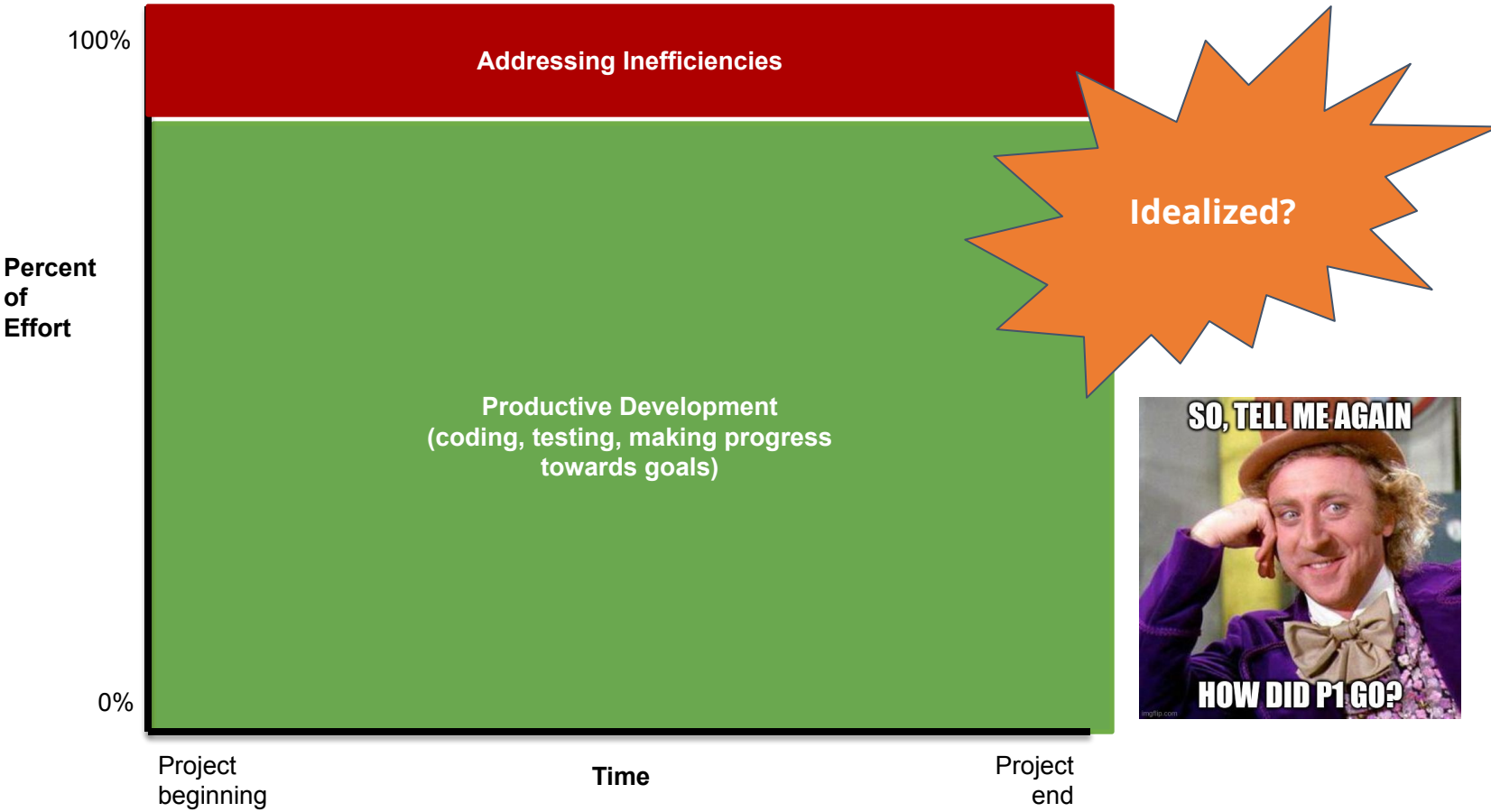
Test the code to identify the defects

Debug to find causes of defects

Fix the defects


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

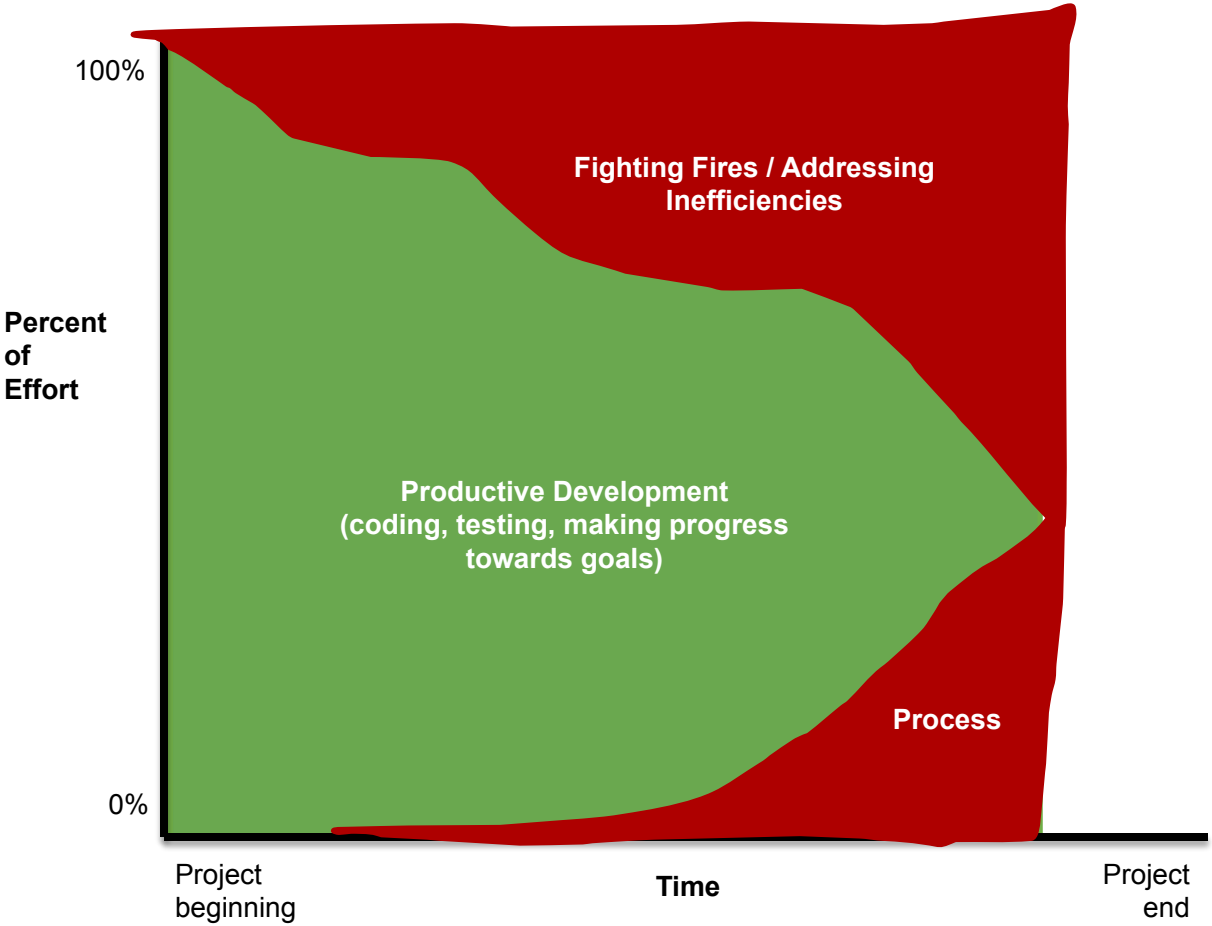
Idealized?


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

SO, TELL ME AGAIN
HOW DID P1 GO?

What happens when …

- Mid-project informal agreement to changes suggested by customer or manager. Project scope expands 25-50%
- Late detection of requirements and design issues.
- Bug reports collected informally, forgotten
- Integration of independently developed components at the very end of the project. Interfaces out of sync.
- Accidentally overwritten changes, lost work.
- When project is behind, developers are asked weekly for new estimates.

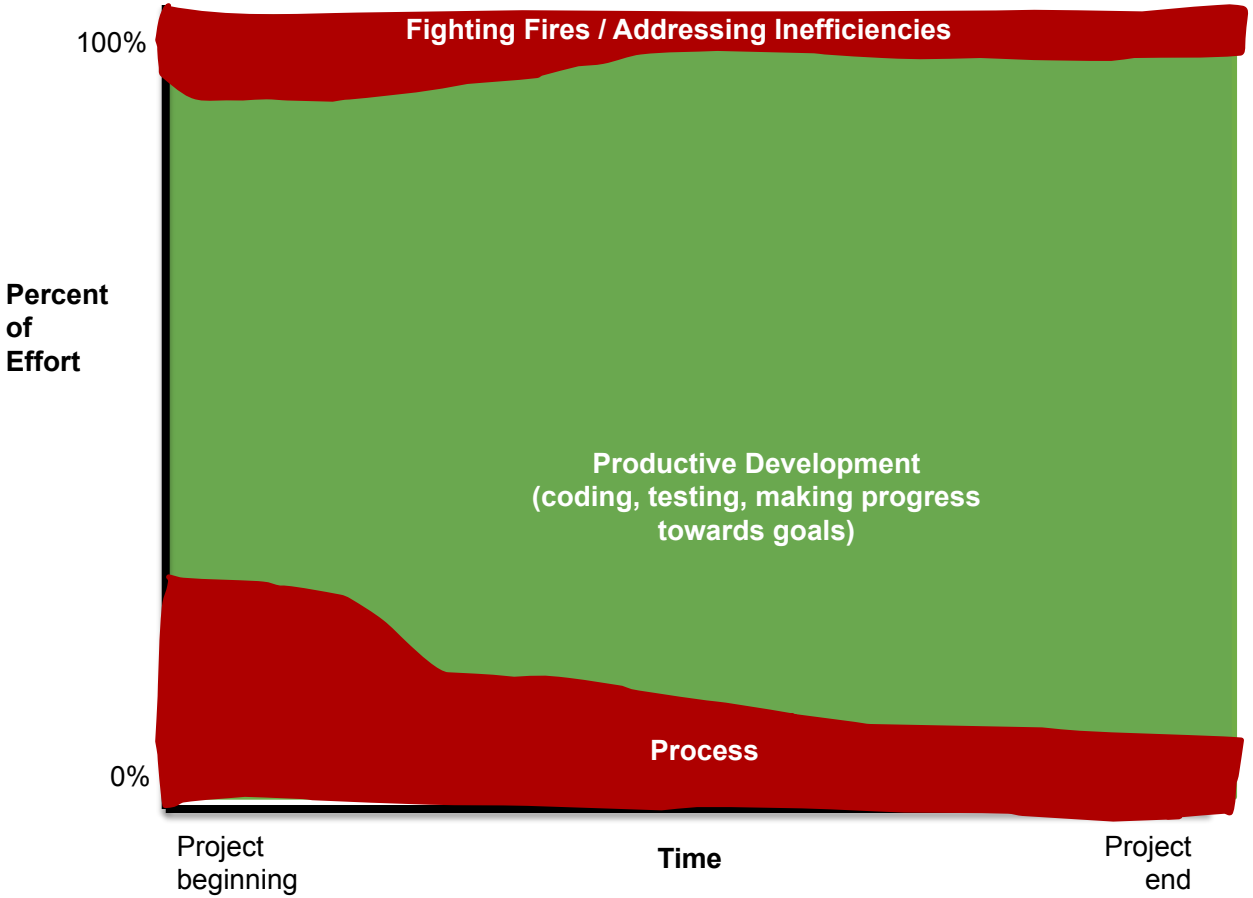

100%
Percent of Effort
0%
Fighting Fires / Addressing Inefficiencies
Productive Development (coding, testing, making progress towards goals)
Process
Project beginning
Time
Project end

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

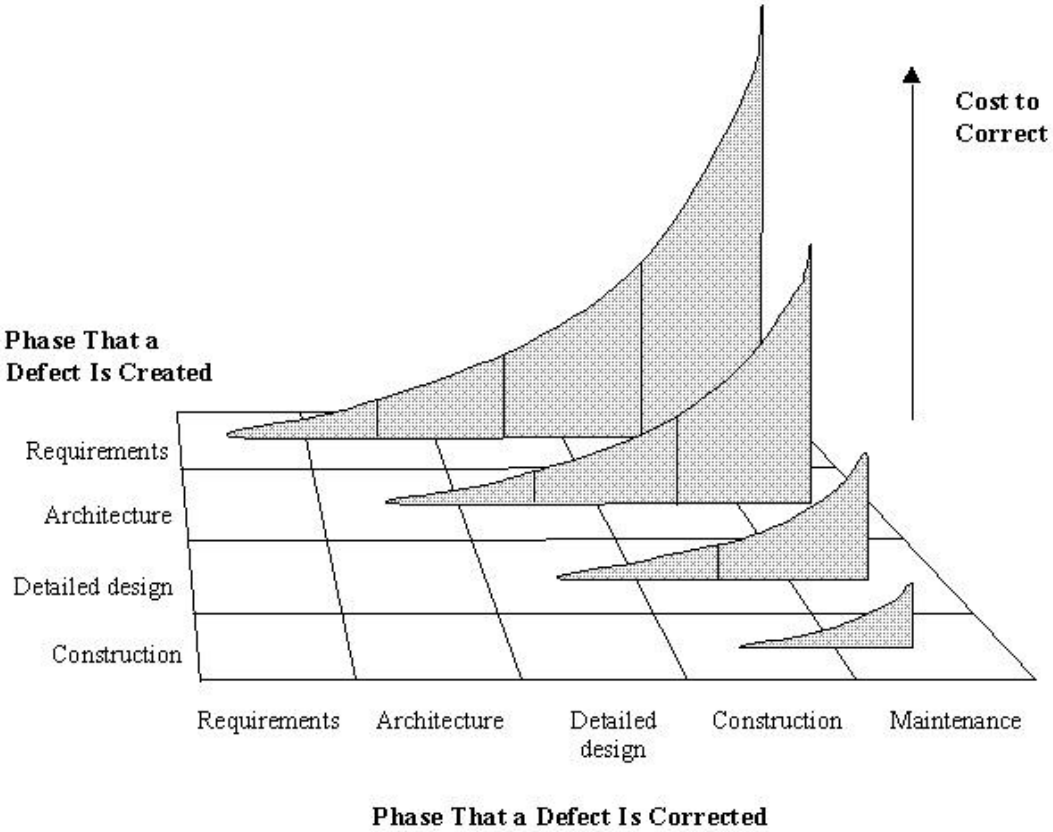

100%
Addressing Inefficiencies
Percent of Effort
Productive Development (coding, testing, making progress towards goals)
Process: Cost and Time estimates, Writing Requirements, Design, Change Management, Quality Assurance Plan, Version Control, Development and Integration Plan, Status Meetings
0%
Project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

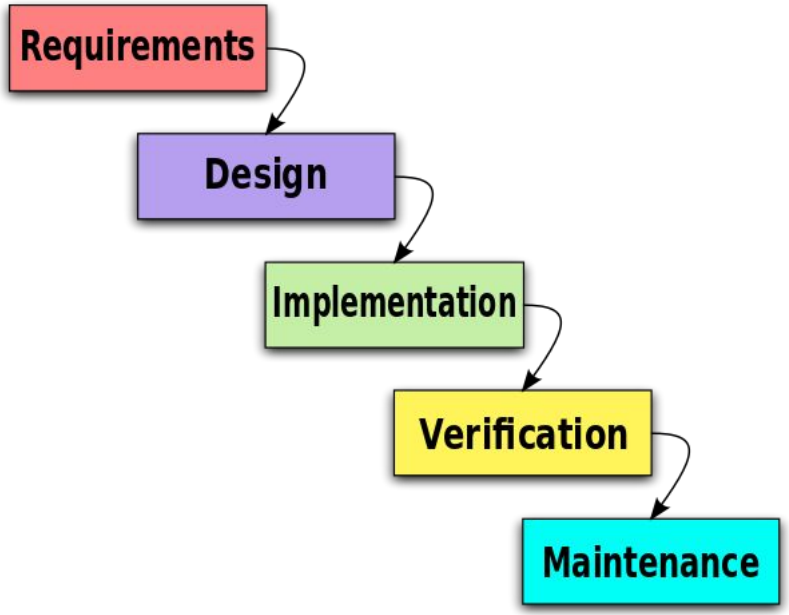
Ideal Curve: Upfront investment for later greater returns



Outline

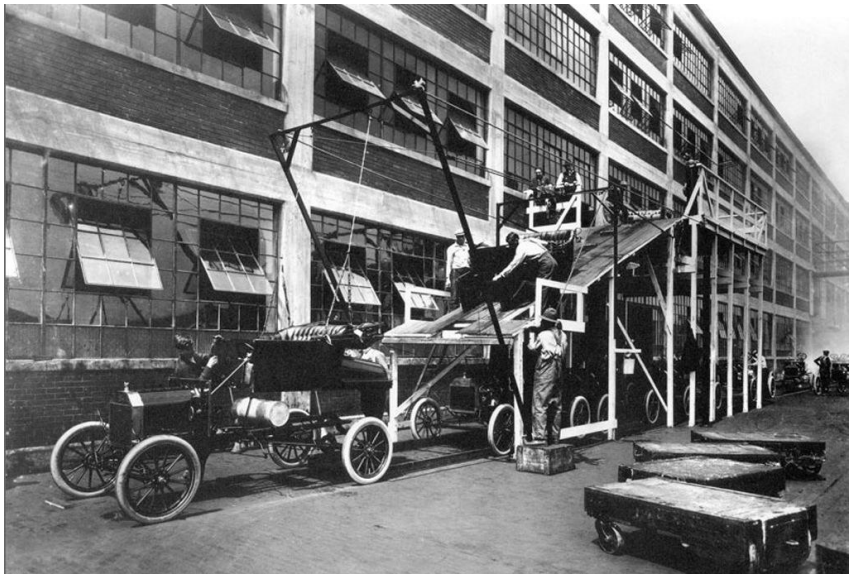
- Software Processes and why we need them
- **Software Process Models**
  - **Agile and Scrum**
- Planning: Task and progress estimation

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) Late 1940s

Build only what is needed, only when it is needed.

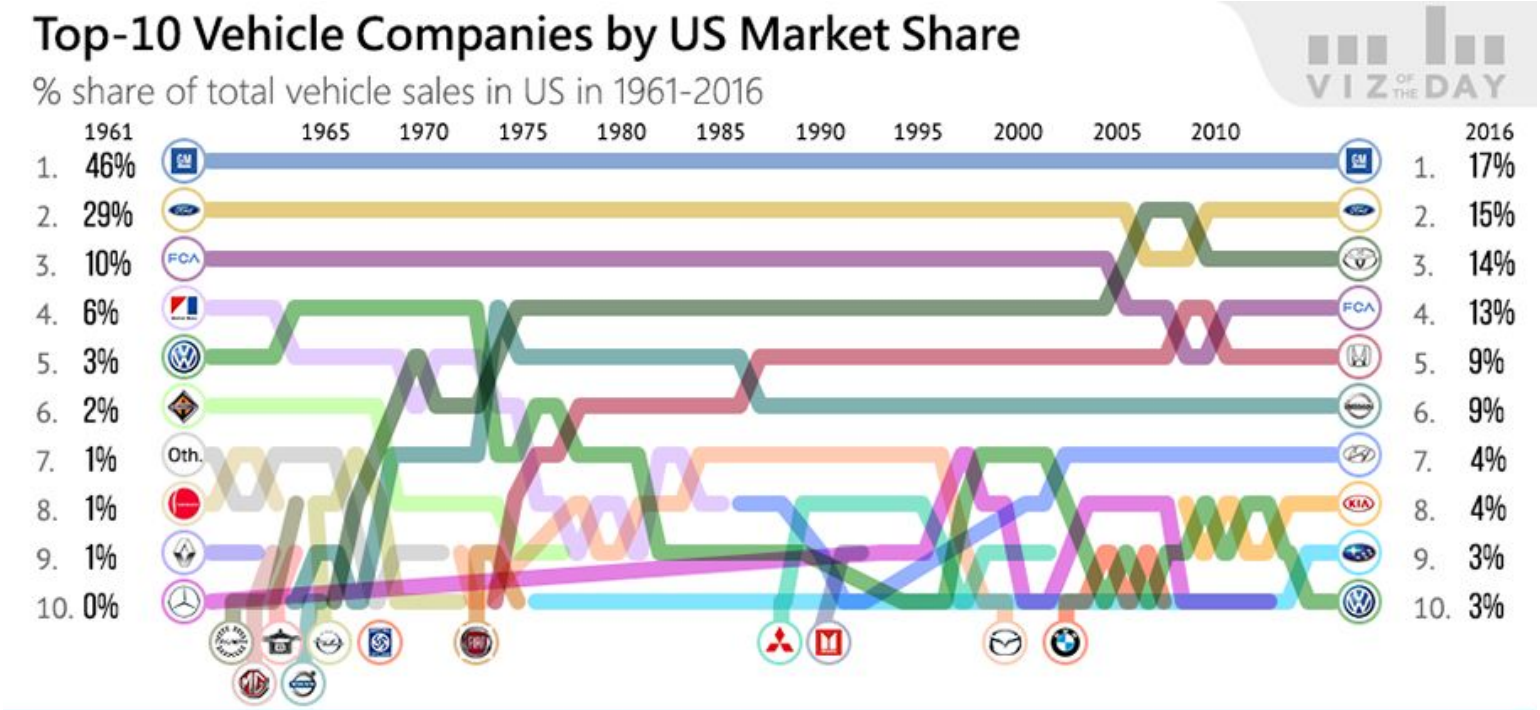
Use the “pull” system to avoid overproduction. (Kanban)

Stop to fix problems, to get quality right from the start (Jidoka)

Workers are multi-skilled and understand the whole process; take ownership

Enabling teams to have autonomy and control to change/improve quickly/continuous improvement (kaizen)

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



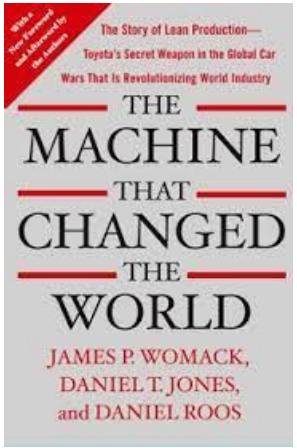
From TPS to Agile

The new new product development game

Stop running the relay race and take up rugby

Hirotaka Takeuchi and Ikujiro Nonaka

1986

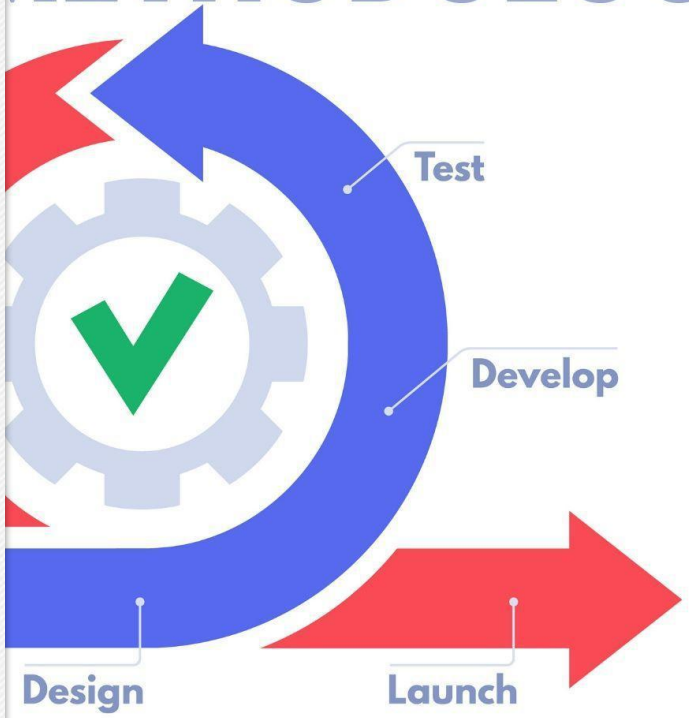


1990

...

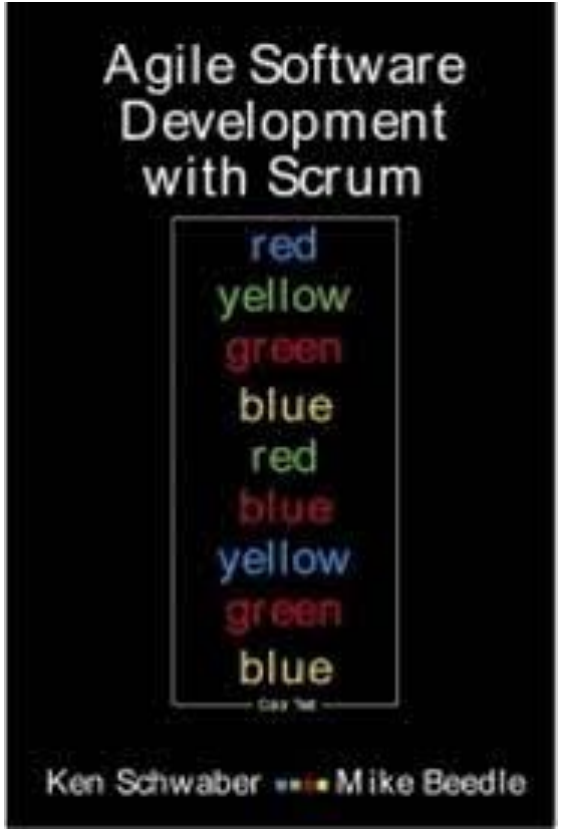
2001

METHODOLOGY

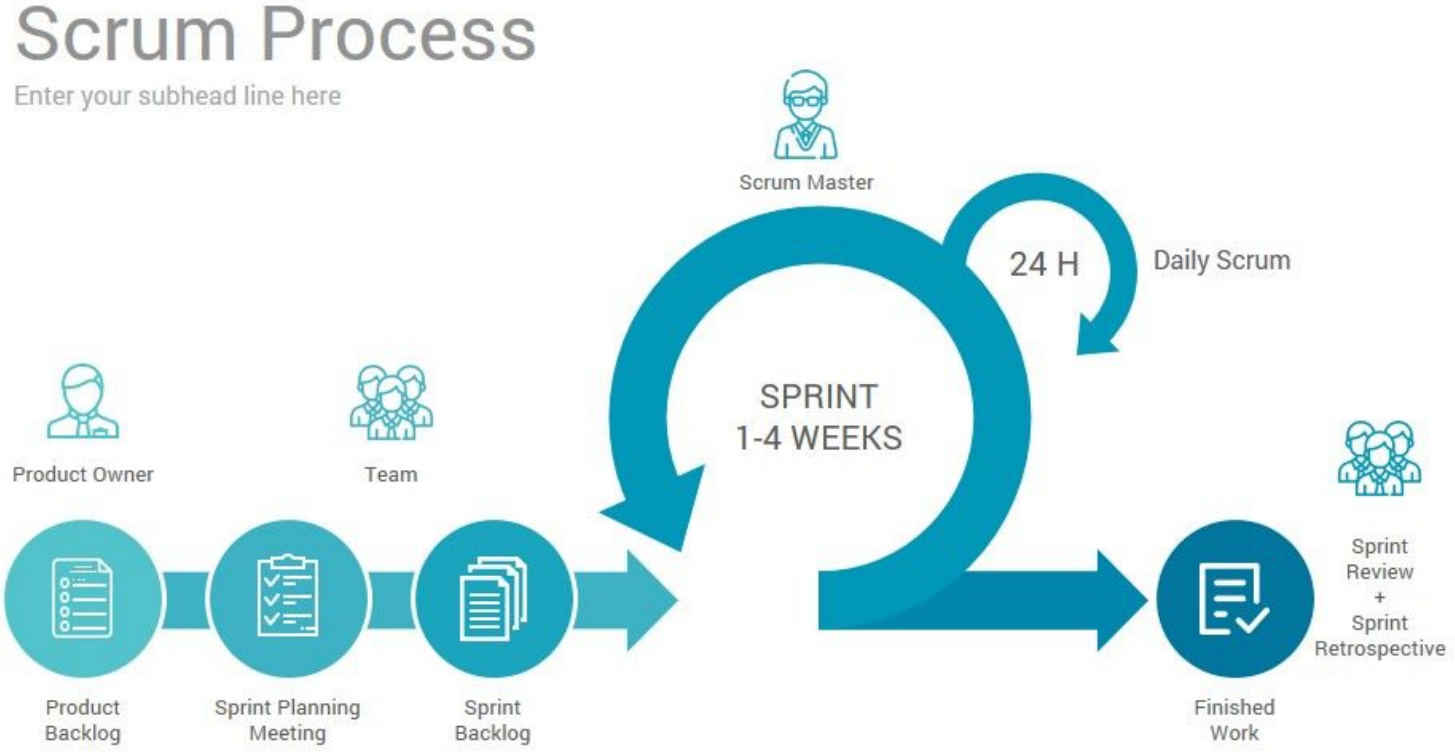


Scrum

(Only a brief intro)



Elements of Scrum



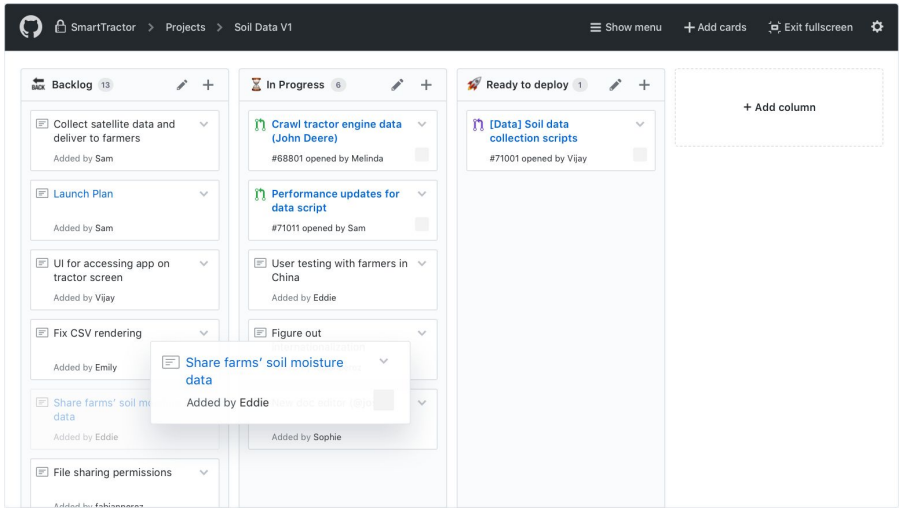
Backlogs

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



Scrum Meetings

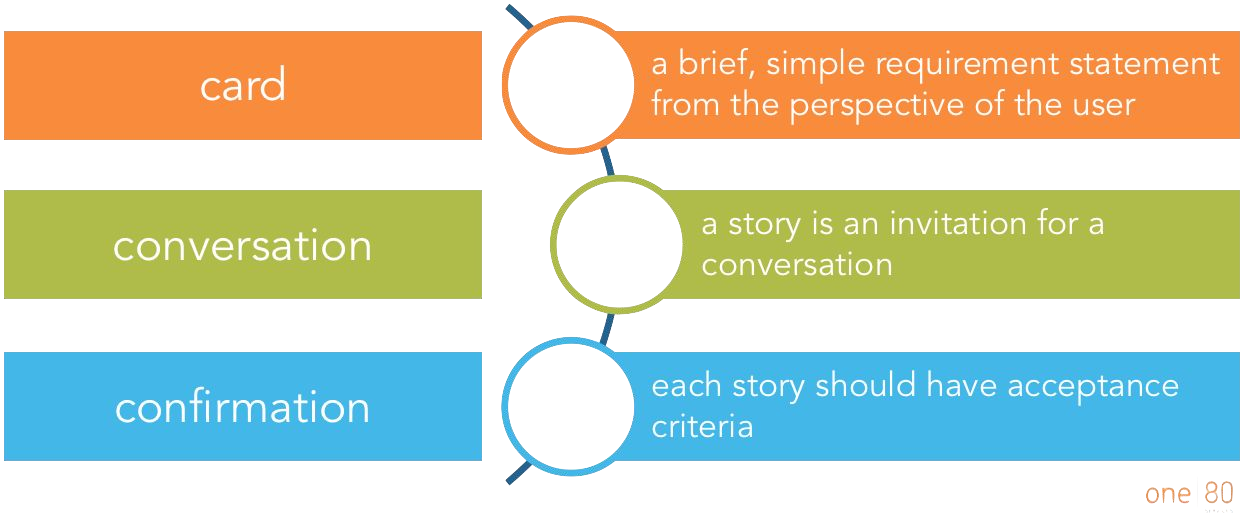
Sprint Planning Meeting
- Entire Team decides together what to tackle for that sprint

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



User story cards (3”x5”)

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

Conversation

- 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?

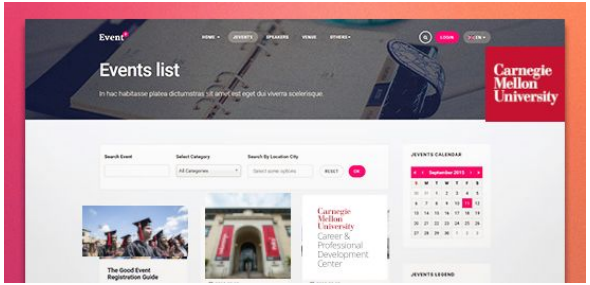
Follow the INVEST guidelines for good user stories!

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

one 80

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

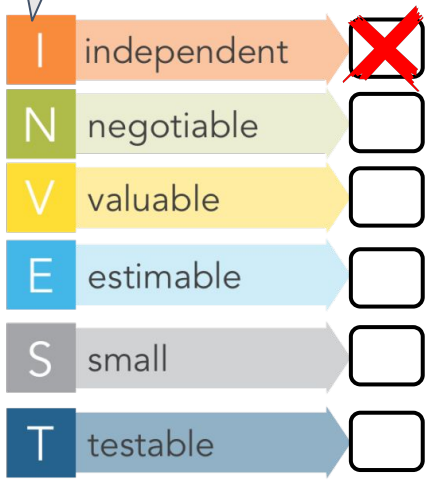
- Schedule in any order.
- Not always possible

Counterexample

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.



Negotiable

- 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!”

Valuable

- This story needs to have value to someone (hopefully the customer)

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.

Estimable

- 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.

Small

- 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.

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.

Activity: Evaluate using INVEST

User Story #1

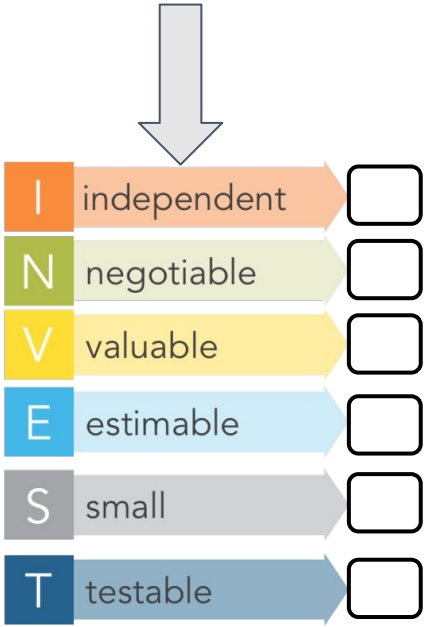
As the Events Coordinator, **I want** the website to seamlessly integrate with various academic calendars and departmental schedules, **so that** event information is always synchronized and accurate.

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?

Select the most serious flaw



User Story #2

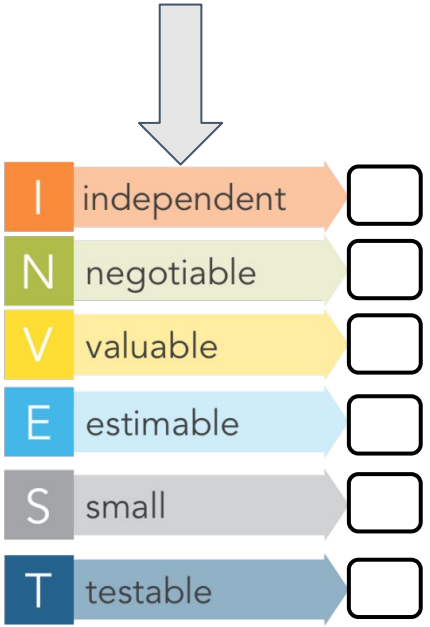
As a student, **I want** the website to have an intuitive navigation system **so that** I can find events effortlessly.

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?

Select the most serious flaw

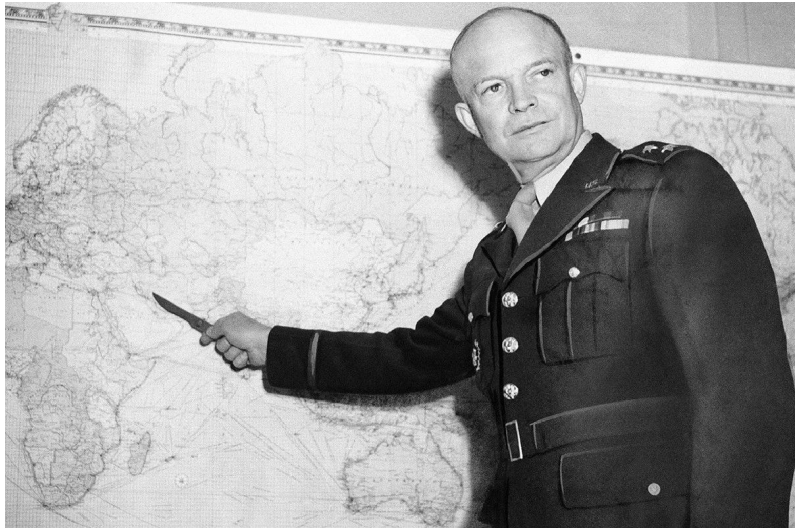


Outline

- Software Process and why we need one
- Software Process Models
- Scrum
- **Task and progress estimation**

Outline

- Software Processes and why we need them
- Software Process Models
  - Agile and Scrum
- **Planning: Task and progress estimation**



“Plans are nothing, planning is everything”

-Dwight D. Eisenhower

Time estimation

https://xkcd.com/612/

Activity: Estimate Time

Enter your answer here:

https://bit.ly/313-estimation

Only the number of days

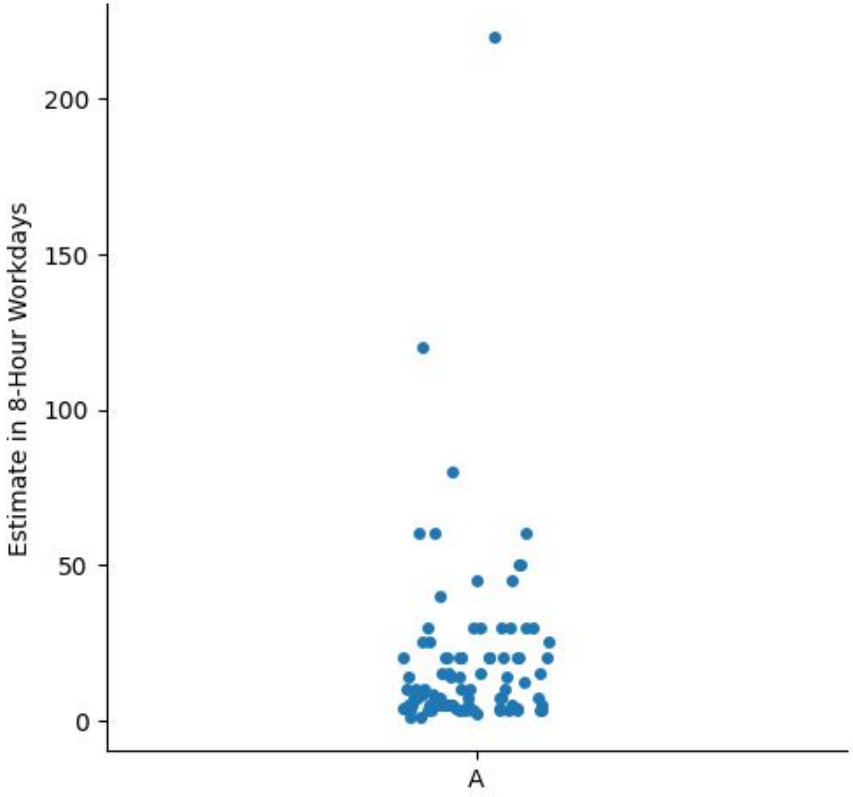
Task A: Simple web version of the Monopoly board game with Pittsburgh street names

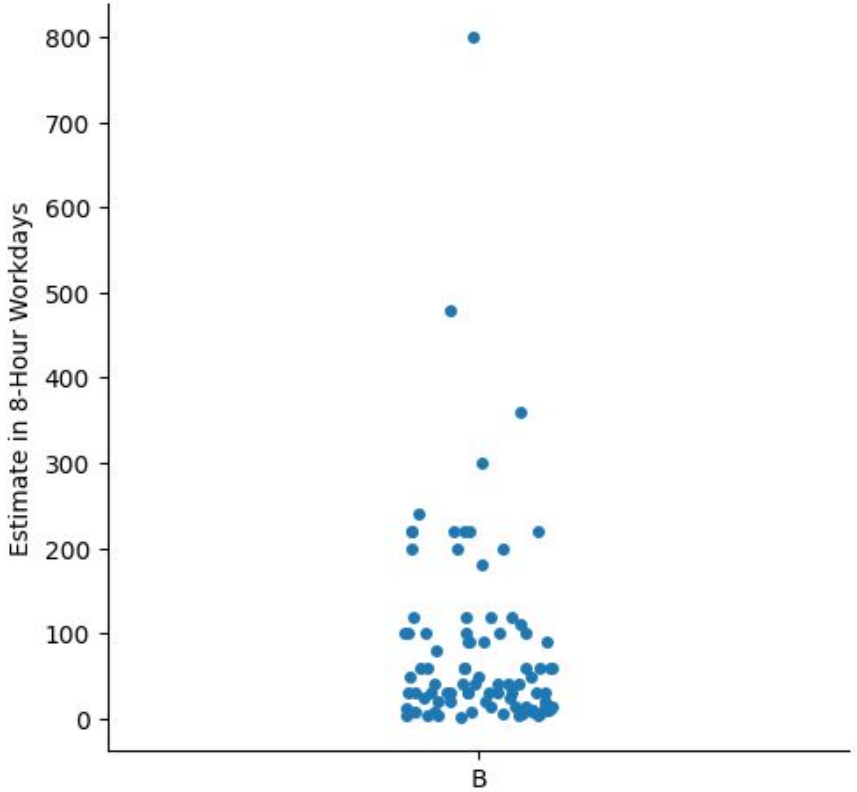
- Developer Team: just you

Task B: Bank smartphone app

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

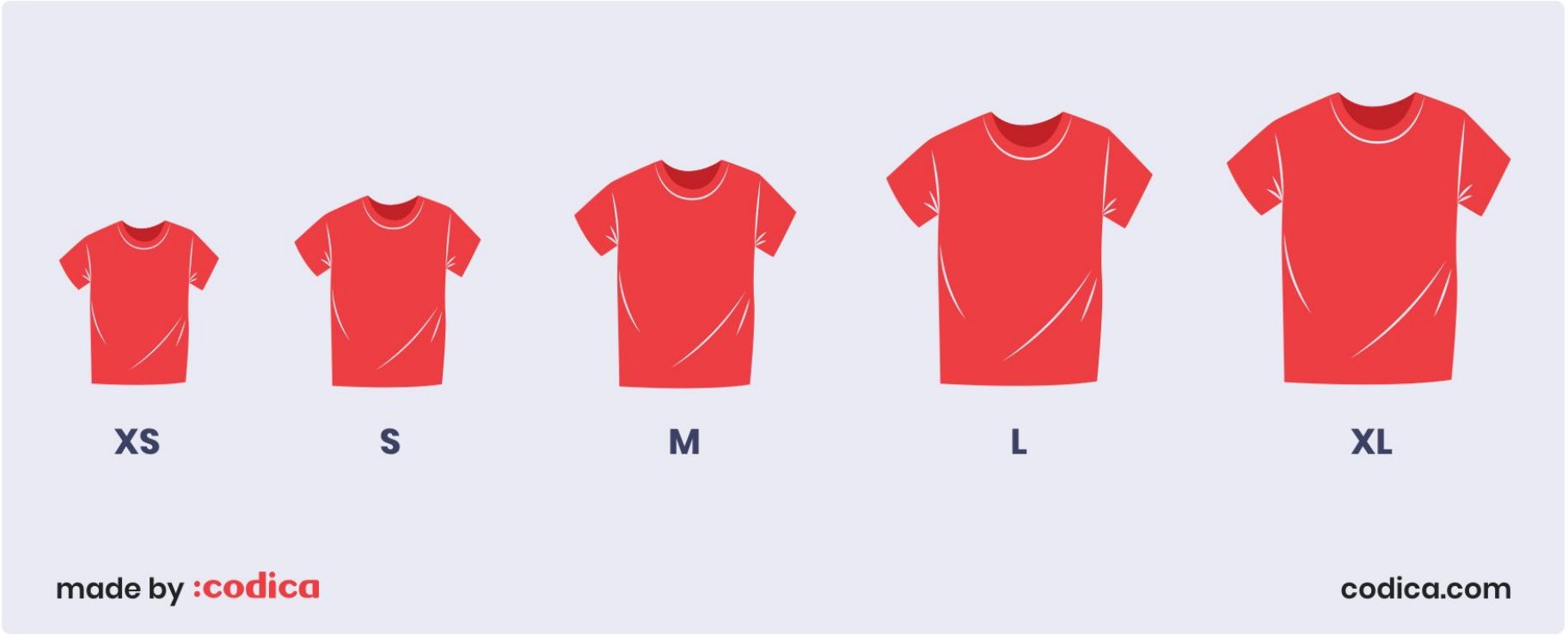
* Estimate in 8h days (20 work days in a month, 220 per year)

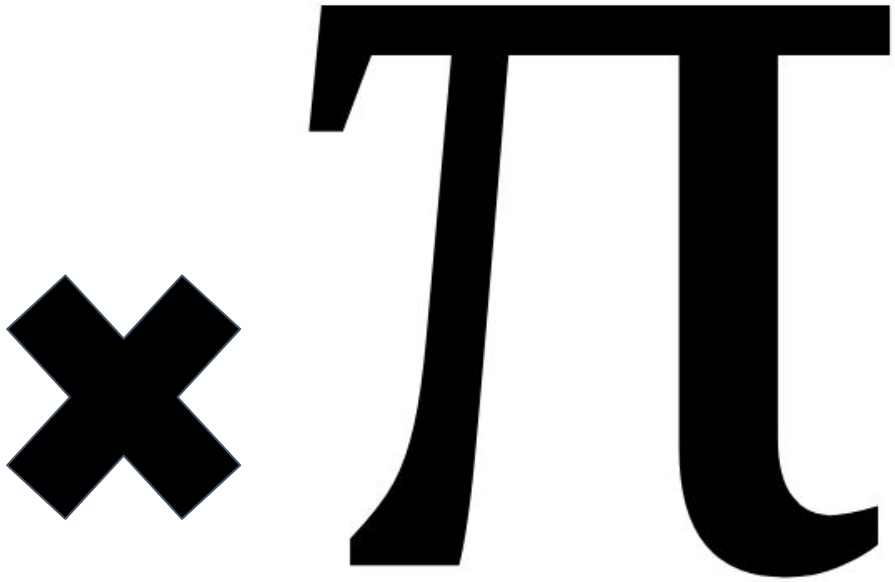

Estimate in 8-Hour Workdays
200
150
100
50
0
A


Estimate in 8-Hour Workdays
800
700
600
500
400
300
200
100
0
B

Improving Time Estimates

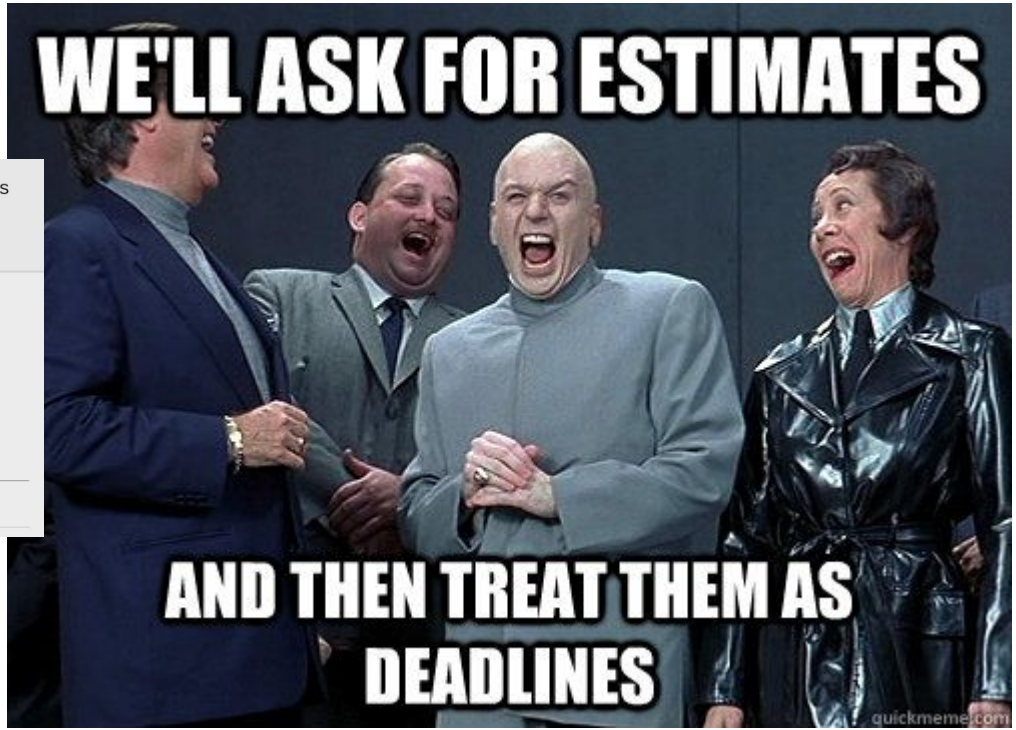
- Prevent conformity bias
- Do you have a comparable experience to base an estimate on?
- How much design do you need for each task?
- Break down the task into smaller tasks and estimate them.


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



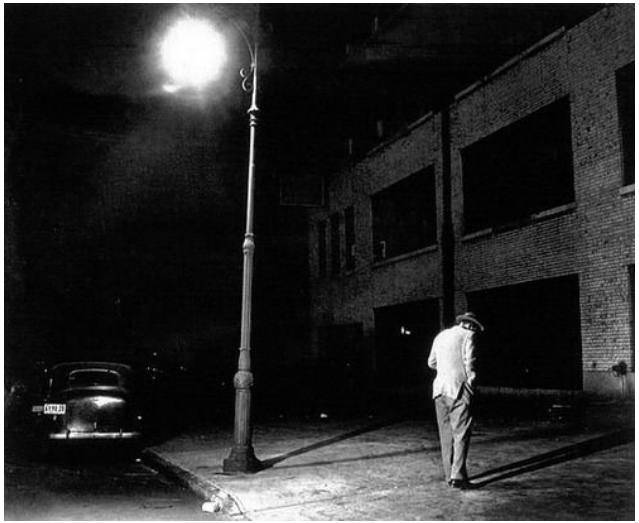
Is Estimation Evil?

Ron Jeffries's essay [Estimation is Evil]



Measuring Progress?

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



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.”

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