W4156

Surviving and Thriving in Software Engineering

Agenda

- ❑ Quick Intros
- ❑ Software is eating the world
- ❑ Software Engineering in Industry
- ❑ Course Outline / Approach
- ❑ Logistics

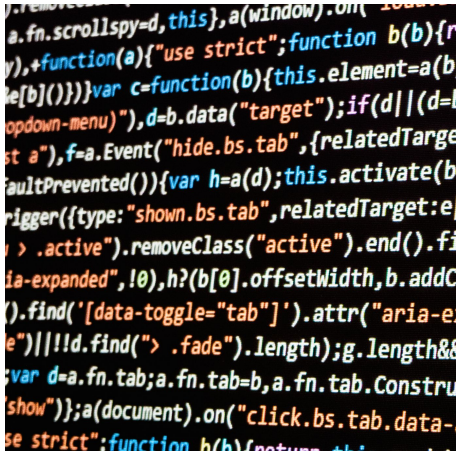
Quick Intros

Who am I and why am I here?

Quick Intros

Ewan Lowe
Executive Director
@ J.P. Morgan

Software Engineer



Unhealthy Obsession

Motivation

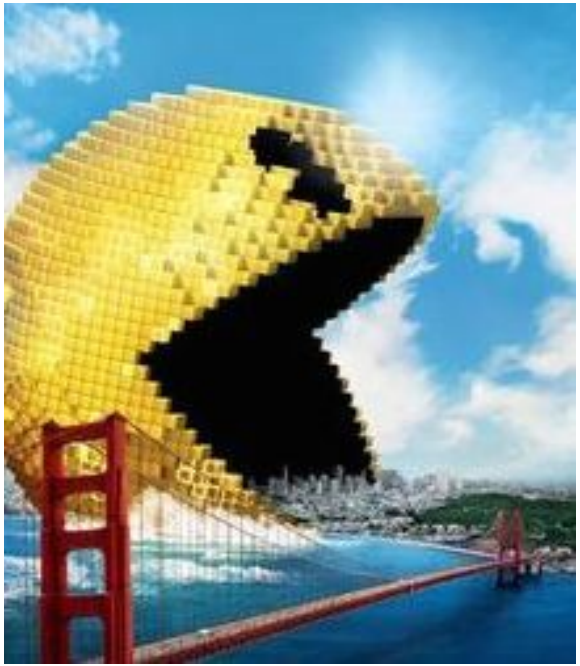
- Took this course N years ago where someone from industry came in to teach
  - Had and continue to have benefit of great mentors throughout
  - Opportunity to give back to engineering community
- Historically, took a grad each year. Teaching course is more systematic solution!
- Enjoy teaching the course ?!?!?
- Made a variety of interesting and dumb mistakes to learn from ….

Software is eating the world

Why being able to ‘code’ is a near mandatory part of a future career

The importance of courses like W4156

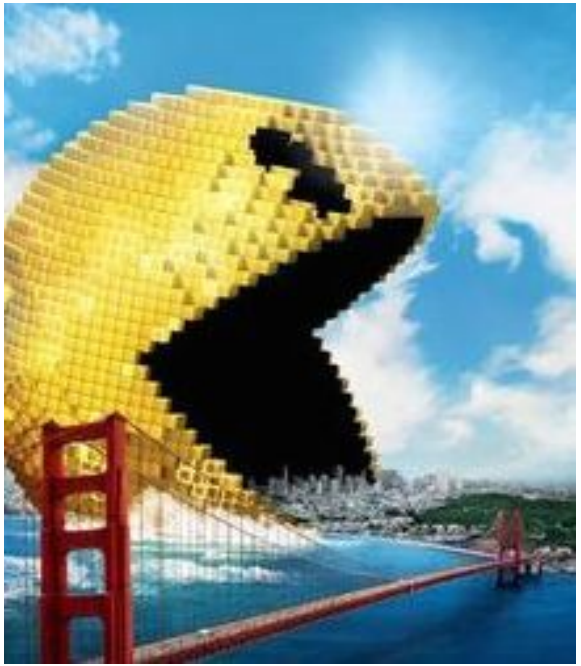
Software is Eating the World



Picture: Pixels the movie

Marc Andressen https://a16z.com/2016/08/20/why-software-is-eating-the-world/

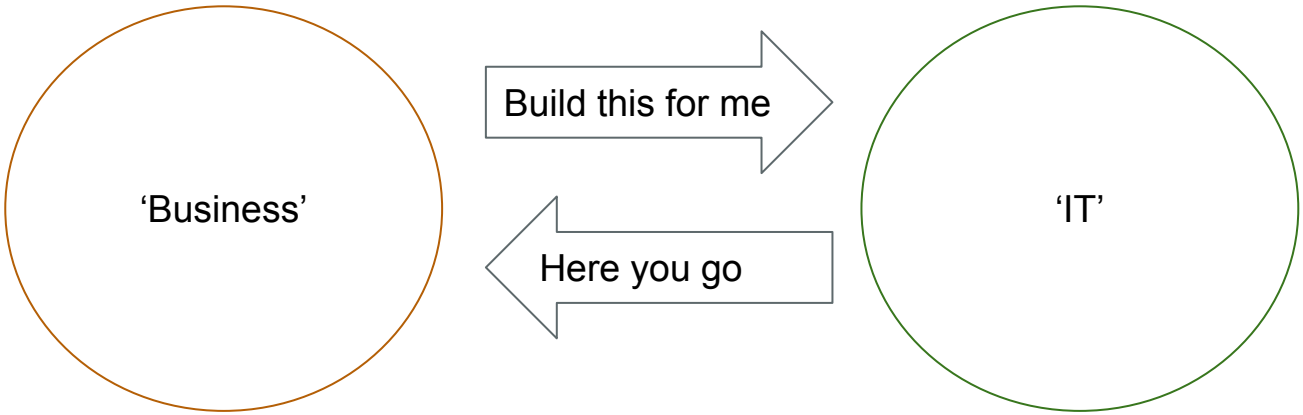
Software is Eating the World



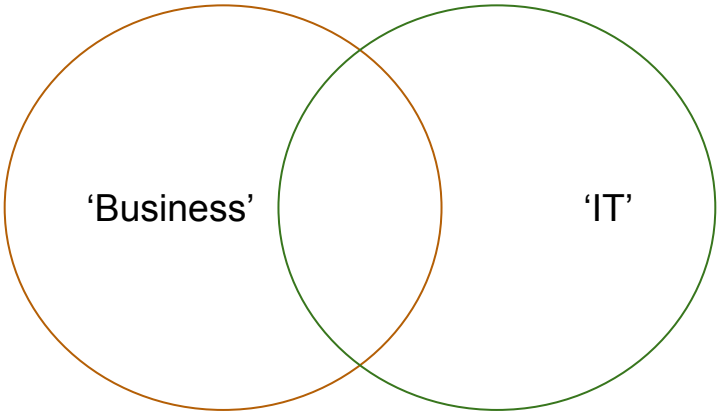
Most jobs need *understanding* of building software

1. Building software (direct)
2. Business users being ‘self service’ (writing queries/scripting)
3. Business users *commissioning* software (how does *process* work?)

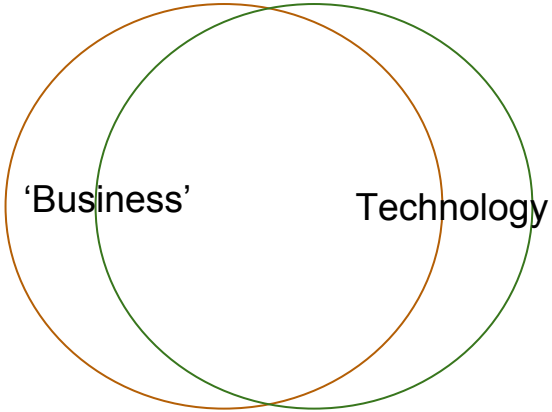
2000



2012



2025



Prediction: We will see the acceleration combined business-technology groups

The business will be built on technology and ‘self service’ teams will move x10 times faster

Software Engineering in Industry

The difference between how we start to learn to ‘program’ vs how it works in industry

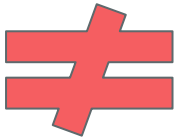
How we all learn to program

How we learn *'programming'**
Well defined (provided problem / write sort etc)
Computer science 'problem domain'
0 existing code
Don't use existing code / plagiarism
Alone
No 'production'
Small codebase / 4-5 weeks
No / very ad-hoc testing
No hardware failures
Focus on functional requirements

*Perfectly valid way to learn data structures, algorithms and the syntax of programming languages ……

Versus Reality / Industry

How we learn ‘programming’
~~Well defined (provided problem)~~
~~Computer science ‘problem domain’~~
~~0 existing code~~
~~Don’t use existing code / plagiarism~~
~~Alone~~
~~No ‘production’~~
~~Small codebase / 4-5 weeks~~
~~No testing~~
~~No hardware failures~~
~~Focus on functional requirements~~



Software Engineering in Industry
Users! Define problem. Extract requirements!?!
Hotels, Finance, ***Shellfish***, Furniture, …..
Mostly ***‘brownfield’*** / existing code
Use ***anything*** that cuts time + ***integrate***
In ***teams*** (or in *organizations*)
Constantly ***running*** 24x7
M-Bn LOC to be maintained for ***years***
Testing, Testing, Testing *and more* ***Testing***
Real world is ***fragile***
Availability, Performance, Security, Cost, ... , ...?!?!

Uh Oh

(I gave this as a guest lecture in fall and at this point you could hear a pin drop)

W4156 Industry Skills

What are the fundamentals and experience we need to thrive in industry?

Course Goals

To ‘bridge the gap’ and equip students with;

- the ability to take an idea from inception through to working software product
- an understanding of the theoretical underpinnings of software engineering (‘*why*’ the practices)
- practical exposure to the software engineering lifecycle (processes and tools)

For those students who choose to pursue a career in software engineering the course will cut their time to achieve basic competency and be able to contribute to a team in industry

Course Outline

Software Engineers from Industry
Panel Discussion

Lecture Track

Product Requirements → Testing/ TDD → SCM, CI → Methodology → OO AD → Good Code → REST → Architecture

</> Code by 3rd week

Applied in project (as close as possible)

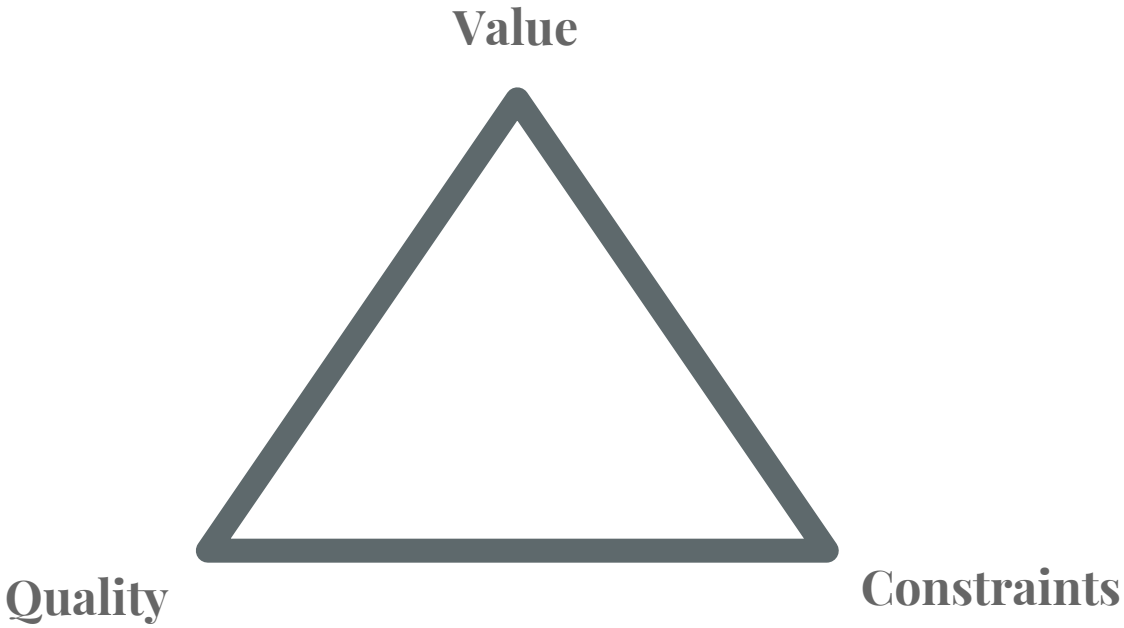
Project

+ Mini-milestones

Form Teams → Product Vision → Sprint 1 → Sprint 2 → Sprint 3 → Demo Day

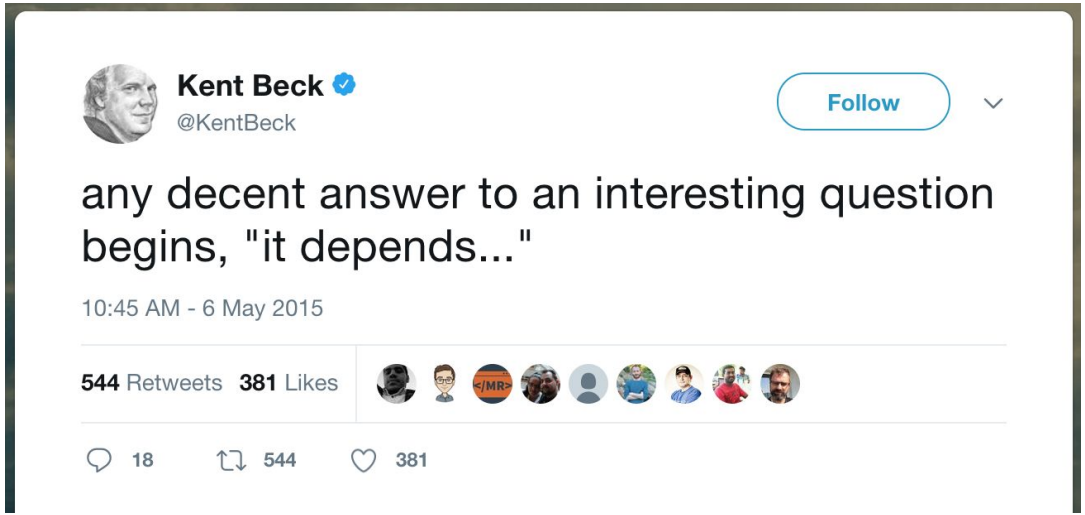
(Challenge of teaching W4156 is it is not possible to cover ‘everything’ by start of coding)

Theme I

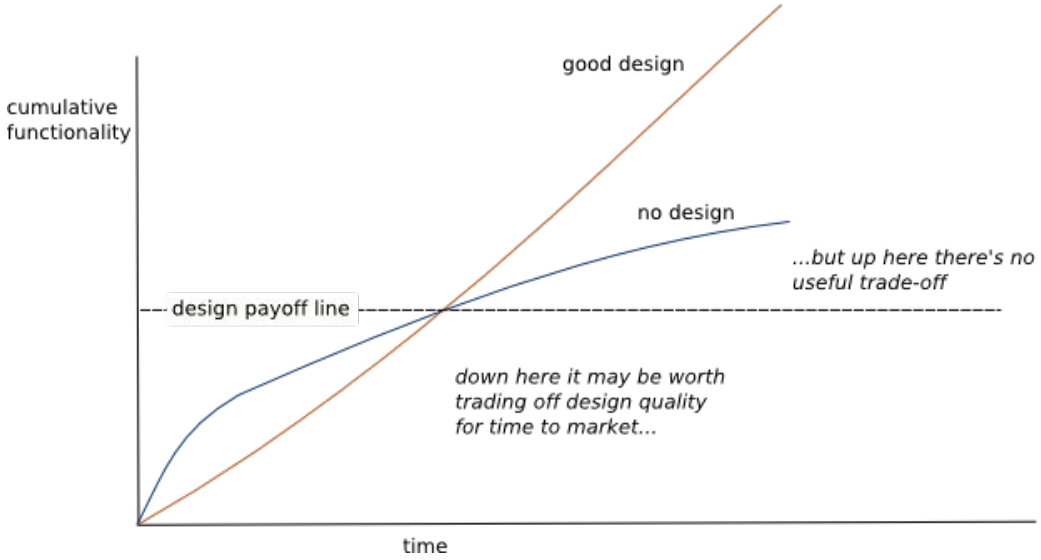


You may hear ‘on time’, ‘on budget’ and ‘agreed scope’. We will discuss in a latter lecture this is a different perspective than above

Theme 2


Kent Beck
@KentBeck
Follow
any decent answer to an interesting question begins, "it depends..."
10:45 AM - 6 May 2015
544 Retweets 381 Likes
18
544
381

Theme 3

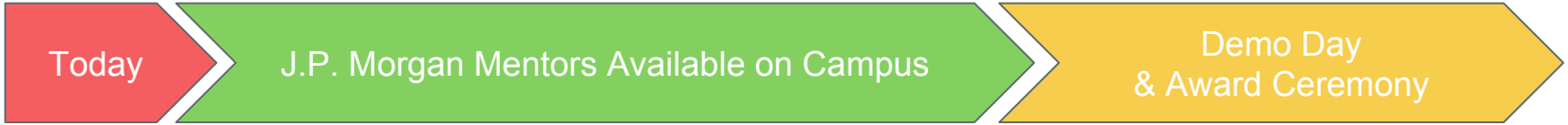


https://martinfowler.com/bliki/DesignStaminaHypothesis.html

The Project

1. Teams of 4 (teams form after change grace period)
2. Build something you devise (I can help provide project ideas if required)
3. Homework milestones connect lectures with project (produce a PRD, planning, etc)
4. Optional: I have secured $150 of AWS credits for teams to use this term if desired
5. Demo Day: Demo your project to engineers and senior management on site at the JPMC Campus in Midtown, prizes (and potentially swag)

Project JPMC Partnership



Once teams form:

- 6-7 experienced engineers available
- Be here for blocks of time on campus
- Help applying lectures to project

(But not allowed to code the project!)

- Class comes to to 383 Madison
- Demo to JPM employees
- Judging panel of senior executives or engineers
- Award ceremony (independent of course grading)
- Drinks, Canapes and hopefully a unique view over Manhattan!

What this course is not

1. UX/UI Course: not enough time to teach everything (and I will not teach react, angular etc)
2. Coding Bootcamp: Interweave theory and practice
3. Entrepreneurship/ Startup Incubator:
    a. Though many skills are *constituents* - product vision, agile, engineering, etc
    b. If you *want* to course to work on idea - superb! (Though grading criteria the same)
4. Advanced engineering
    a. Course title is ‘ASE’ but it is really ‘BSE’
    b. We could spend 10 weeks on any *one* of the topics we touch
5. Everything you will ever need to know. This course is basic training, map, parachute and rations

Course Approach

- Incremental
  - Many topics are inherently linked / difficult to teach in linear way (e.g. testing <-> design)
  - Will teach topic A then link later (there will be a wonderful moment when all topics link)
- Learn by doing
  - Will provide a skeleton project and refer in lectures. Students can use project or diverge.
    (Docker, Python 3.6, Flask, Swagger, AWS CI/CD pipeline and deployment, AWS Credits)
- Foundations
  - Millions of different technologies and methodologies. Technologies evolve quickly
  - Will teach ‘a’ methodology but equip you with the skills to understand/adopt/challenge others
- Judgement
  - Many situations require engineering judgement
  - “It depends” will be a frequent answer but we will pursue “depends on what”
- *Commercial* perspective:
  - Software is generally produced for a business. Process is therefore time, cost and risk aware

Logistics

- **Prerequisites:** Confident ability to program
- **Grading**
  - Exam - 25% midterm, 25% final.
    - Combination of core concepts plus application of concepts
    - Multiple choice plus brief answer
    - Grasp of concepts not rote definitions
  - Project: 50% with 10% group grade
  - Curve: Yes (as far as I can understand all courses are to some extent curved)
- **Room:** 313 Fayerweather
- **Piazza:** All course announcements
- **Attendance:** Mandatory. Raise exceptions with the TAs. Allowed 2 misses.
- **Technologies:** Set of industry tools (details on github)
- **Lectures:** ***in the folder on piazza*** (***not*** attached to each lecture)

Book

- Books are surprisingly tough for this course.
  - Some are either academic theory (½ of what we need)
  - Others are procedural and lack the ‘why’ / theory (the other ½ we need)
  - Other books focus on only one aspect of the end to end lifecycle:
    - Product / Requirements, Methodology, Testing, Construction, Testing…
  - None really keep up with state of the art (CD, Cloud, etc)

- Response: Core textbook is ‘Beginning Software Engineering ’pulling in other materials as required

Bilateral Contract

- The course is oversubscribed. If you are going to take a slot then utilize well
- Basics: Be punctual. Do the reading. Study the code.
- Work *evenly* throughout the course (make a dent in the project before mid-term)
- We all want a lively class:
  - I have made effort to mix up lecturing, interactive sessions and code
  - Interrupt, Challenge, Clarify
- This is the second year (and first year tweaking material)
  - Give me feedback

Classmate quotes once Internships start ....

“Throughout the semester, whenever I told my best friend (who currently works as a software engineer) about the class and the topics we were covering, he always said he wished he had taken the class because such topics were already relevant to his firsthand experience in industry [...]. Now I fully understand and appreciate why.”

“[after I started my internship was] when I realized how valuable the specific material you chose to teach us was. That’s when I realized some of my peers — the ones who had previously laughed off [the lectures on] *process* rather than *coding* — didn’t understand they've never been in industry full-time. You taught us what being a software engineer in real life will be like, regardless of our naive expectations.”

“This was by far the most practical and useful class I've taken [...]. I've been helping someone on a project and am using a lot of what I learned to help the project improve (ie. building out a test suite since they currently have 0 tests in place before they deploy their product to users)”

“Just a quick note to let you know that while I'm working on software projects in my internship (read: redoing one 2nd time in a week because my manager came up with a "new plan" aka didn't really think through the initial plan when handing me the project), I've come to the realization that nothing rang more true to me than the slide about the problem domain”

“I was just writing to tell you that I think I have applied a lot of the things I learnt in the Advanced Software Engineering class at my internship and it’s only been 3 weeks!”

Partnership this term

Mentorship and support from some senior software engineers from industry

?