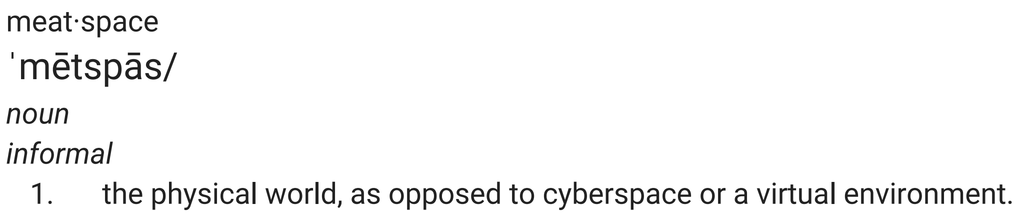
W4156

Meatspace: People & Farewell

Prologue: Meatspace

Am pretty confident by now we are fully aware software is made by groups of humans (interacting with each other) often delivered to users who are also human.

This voyage into meatspace pulls in psychology, communication, group dynamics, organizational behavior .....



meat·space

ˈmētspās/

noun

informal

1. the physical world, as opposed to cyberspace or a virtual environment.

Agenda

❑ The Individual: 10x / What makes a great engineer?

❑ The Group: Engineering Culture

❑ The Architecture: People ↞ ↠ Architecture

❑ Final thoughts

What is the productivity disparity / distribution between engineers? (10X)

10x

10x: thesis that the *best* software engineers are order of magnitude more productive than average software engineers

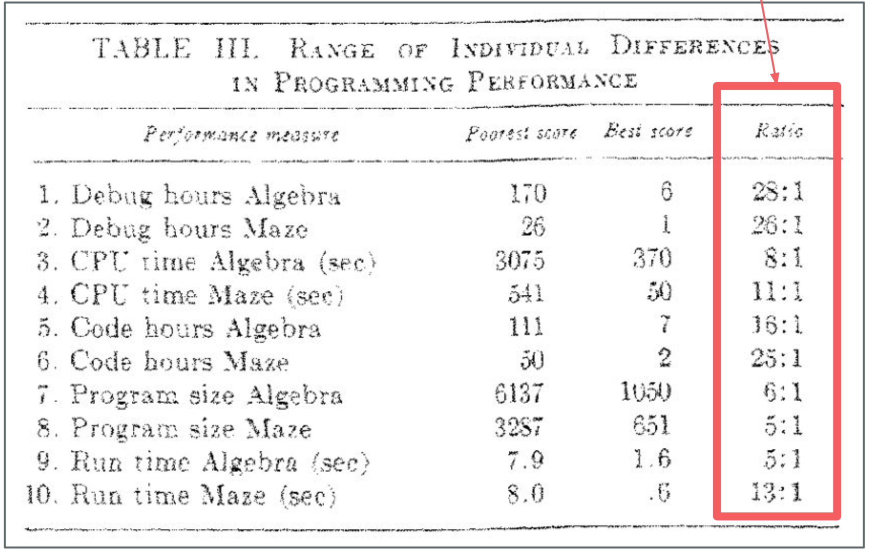


TABLE III. Range of Individual Differences in Programming Performance

Performance measure	Poorest score	Best score	Ratio
1. Debug hours Algebra	170	6	28:1
2. Debug hours Maze	26	1	26:1
3. CPU time Algebra (sec)	3075	370	8:1
4. CPU time Maze (sec)	541	50	11:1
5. Code hours Algebra	111	7	16:1
6. Code hours Maze	50	2	25:1
7. Program size Algebra	6137	1050	6:1
8. Program size Maze	3287	651	5:1
9. Run time Algebra (sec)	7.9	1.6	5:1
10. Run time Maze (sec)	8.0	.6	13:1

Ratio best:worst

Exploratory experimental studies comparing online and offline programming performance

10x is controversial and there have been criticisms of original study. However, it forms a contentious idea around which to explore our question

Can 10X be true?

Creative work *is* different than *mechanical labor*

- 10x engineers can shorten path/meander
  - Not about *rate of work* but *much less work*
  - Can bypass two ‘design generations’/’rewrites’
  - ‘Good’ decisions *compound* over life of project
- Mentor team of N (N engineers * M% productivity boost)
- Not just *productivity*. Can *succeed* in solve problems where others will *fail*
- Sharing libraries provides *leverage* (my code to 100 engineers etc)

10x Results

Classical '10X engineer studies are controlled experiments

- Engineers looking at new/abstract problems
- Controlling for experience

Building a team in industry you care about ***results*** not 'fair comparison/controls'

- Hire someone who has ***seen*** the same problem domain
- Hire ***experience*** (*seen many patterns / problems)*
- Fewer people == less communication and handoff
- Hire people who elevate others

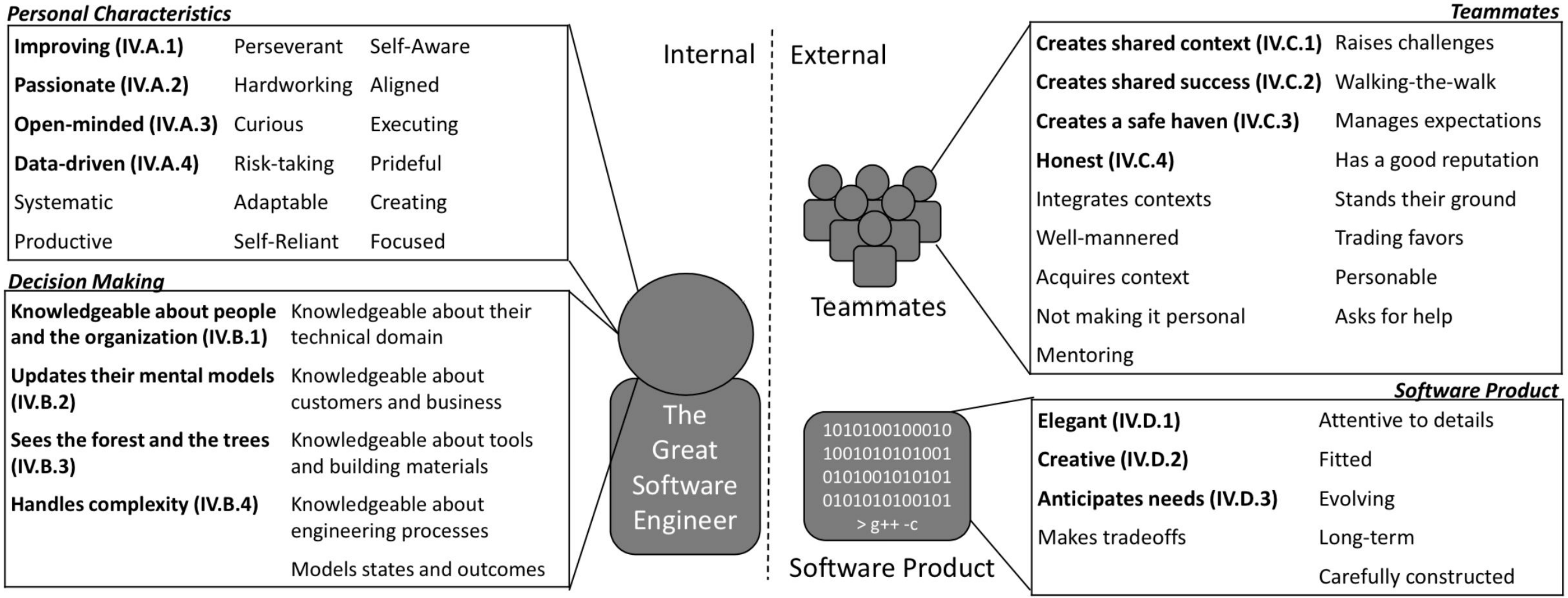
Imagine building a similar project now vs 10 wks ago. How much quicker are you?

Empirically/IMO: '10x results' is true & significant disparity output between S.E.s

What makes a great engineer?

(much broader definition than 10x)

What makes a great engineer?



Personal Characteristics

- **Improving (IV.A.1)**
- **Passionate (IV.A.2)**
- **Open-minded (IV.A.3)**
- **Data-driven (IV.A.4)**
- Systematic
- Productive
- Perseverant
- Hardworking
- Curious
- Risk-taking
- Adaptable
- Self-Reliant
- Self-Aware
- Aligned
- Executing
- Prideful
- Creating
- Focused

Decision Making

- **Knowledgeable about people and the organization (IV.B.1)**
- **Updates their mental models (IV.B.2)**
- **Sees the forest and the trees (IV.B.3)**
- **Handles complexity (IV.B.4)**
- Knowledgeable about their technical domain
- Knowledgeable about customers and business
- Knowledgeable about tools and building materials
- Knowledgeable about engineering processes
- Models states and outcomes

Teammates

- **Creates shared context (IV.C.1)**
- **Creates shared success (IV.C.2)**
- **Creates a safe haven (IV.C.3)**
- **Honest (IV.C.4)**
- Integrates contexts
- Well-mannered
- Acquires context
- Not making it personal
- Mentoring
- Raises challenges
- Walking-the-walk
- Manages expectations
- Has a good reputation
- Stands their ground
- Trading favors
- Personable
- Asks for help

Software Product

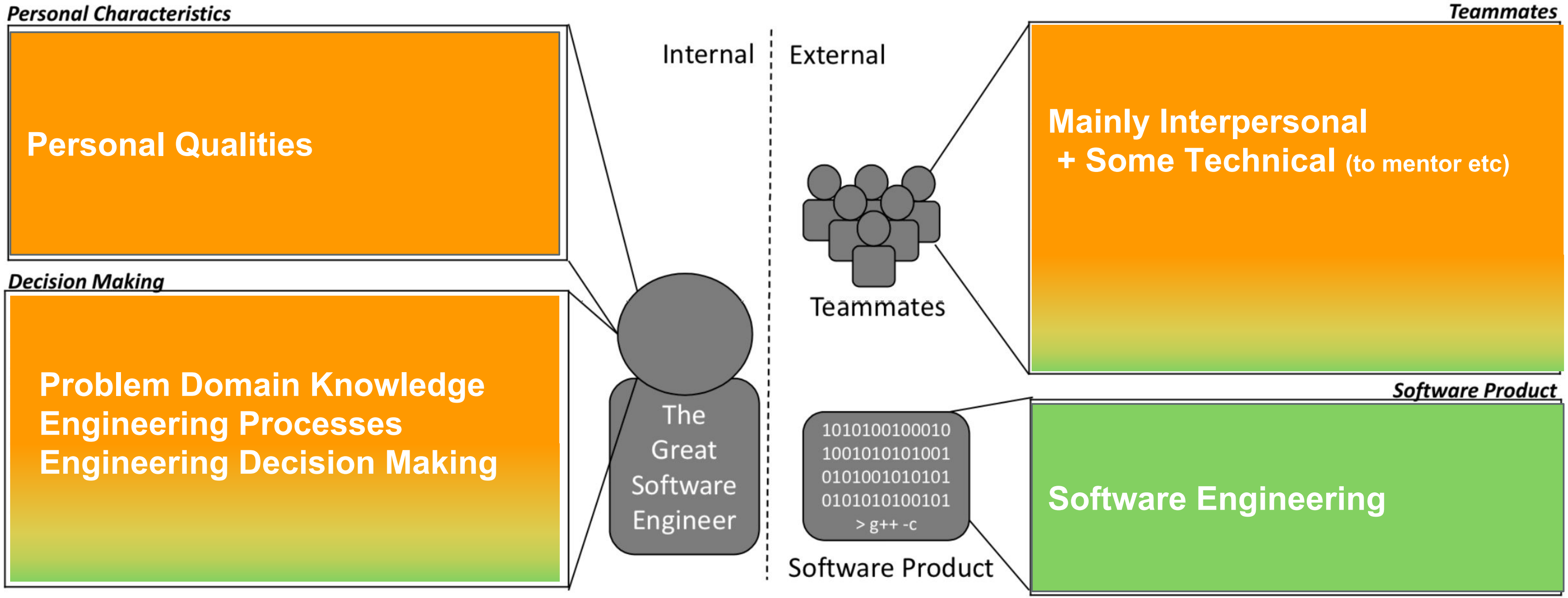
- **Elegant (IV.D.1)**
- **Creative (IV.D.2)**
- **Anticipates needs (IV.D.3)**
- Makes tradeoffs
- Attentive to details
- Fitted
- Evolving
- Long-term
- Carefully constructed

What makes a great engineer?

What makes a great engineer?

🟧 ‘Non-Coding’

🟩 ‘Coding’



Personal Characteristics

Personal Qualities

Decision Making

Problem Domain Knowledge
Engineering Processes
Engineering Decision Making

Internal

External

The Great Software Engineer

Teammates

Teammates

Mainly Interpersonal
+ Some Technical (to mentor etc)

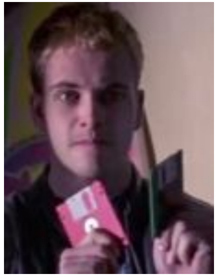
Software Product

Software Product

Software Engineering

What makes a great engineer?

Who would you hire to join a real world team? (*without* special powers)

							
Software Product							
Team Mates (interaction)							
Decision Making							
Personal Skills							

Who would you hire to join a real world team?
(*without* special powers)

Software Product	✓	✓	✗	✓	✓	✓	✓
Team Mates (interaction)	✗	½	✗	✓	✓	✓	✗
Decision Making	½	✗	✗	✓	✗	✗	✓
Personal Skills	½	✗	✗	✓	½	¾	¾
Bonus: How would you manage?							

Aside

MANAGING the UNMANAGEABLE
Rules, Tools, and Insights for Managing Software People and Teams
Mickey W. Mantle | Ron Lichty
Herding Cats: A Primer for Programmers Who Lead Programmers
Guide your programming team to success by becoming a leader, not just a manager
Identify and overcome obstacles before they impact you or your team
Learn from an author with over 30 years of experience in science, engineering, programming, and the management of technical teams
Apress
Managing Humans
Biting and Humorous Tales of a Software Engineering Manager
Third Edition
Michael Lopp
Apress
Peopleware
Productive Projects and Teams
Third Edition
O'REILLY
"The field has needed a book like this for a long time." –Peter Norvig, Google
Debugging Teams
Better Productivity Through Collaboration
Brian Fitzpatrick & Ben Collins-Sussman

What makes a great engineering organization?

Defining

Many dictionary definitions (norms, societal beliefs,) and other definitions

For our purposes:

Culture: "How things *actually* work around here"

- Which *goals* are pursued / what is the mission?
- *How* are decisions made and by *who*?
- Which values do the organization prioritize in decisions?
- Which behaviors and actions are rewarded vs punished?

Describing Cultures

Power Concentration

Approach to Risk / Failure

Individual vs Collectivism

Time Horizon Orientation

Frugality

Freedom

Process

Individual Accountability

Innovation

Reward/Decision ?ocracy

Work / Life Balance

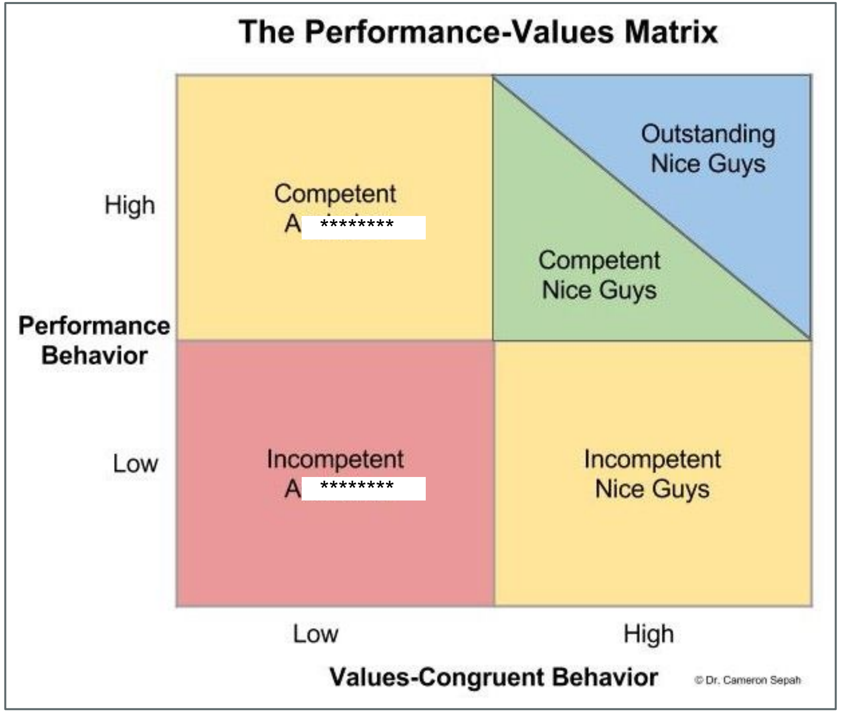
Transparency & Candor

Collection of dimensions drawn from multiple models including OCP, Hofstede

Think culture is fluffy? Consider the different cultures between: prison, air traffic control, government, bridgewater, AWS, Google, etc

- Many cultural models (OCP, Hofstede, etc).
- May have a strong culture (understood and adhere) or weak (not disseminated or rejected).
- Very likely to have sub-cultures (org units / teams)
- You will be surprised by how much this affects your daily life: How easy is it to ‘get stuff done’?
- Effective (vs stated culture) will impact you

Nice vs Performance



The Performance-Values Matrix

Performance Behavior	Low	High
High	Competent A ********	Outstanding Nice Guys Competent Nice Guys
Low	Incompetent A ********	Incompetent Nice Guys

Values-Congruent Behavior

© Dr. Cameron Sepah

Some companies have a ‘rule’ that they never hire the left column. What is that rule called?

Can we understand that rule vs ‘great engineer’ attributes?

Some companies will tolerate top left

Some companies tolerate bottom right / slow to release

Depends on their ***culture***

Sepah

Great Culture

“may well be the most important document ever to come out of the Valley.”
[in reference to the netflix culture deck]
[Sherly Sandberg]

Netflix do not have the monopoly on great culture but certainly there has been a pilgrimage of other industries to netflix/the valley to understand and attempt to replicate/adapt the positive aspects of this culture. There are also emerging ‘cons’ to certain cultures ...

People and Architecture

Conways Law

> "organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations."
>
> [Melvin Conway]

Remember we also saw in microservices that architecture impacted number and level of communication links

Summary

- Pretty sure we have all overcome the hollywood stereotype of the lone programmer
- Many characteristics of great engineers are not ‘coding’
- Engineering culture of organization is complex and multi-dimensional
- Organizational design and architecture are inter-related

Pop Quiz

Question	Answer
Which member of Silicon Valley would you want on your team?	
There is low/high differentiation in productivity between software engineers?	
What are the key aspects of being an excellent engineer?	
Describe the culture of an organization you have worked/seen? (it can be anything from Columbia through to Scouts/Guides)	
What type of culture do you want to work at?	
Describe the relationship between organizational design and architecture.	

Reading

Reading	Optionality
10X	Required
What makes a great engineer	Required
Spotify Engineering Culture and 2	Required
Spotify Steps	Required
Netflix Culture Deck and current and Stripe	Required
Ray Dahlio	Optional

W4156

Course Wrap Up

From our first lecture …..

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~~

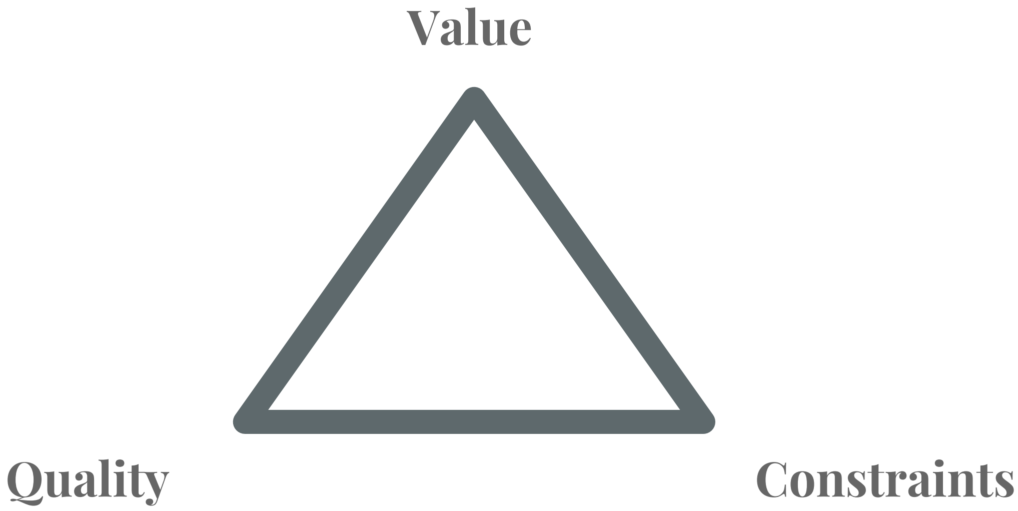
≠

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, ... , ...?!?!

Theme 0: Flow of Value

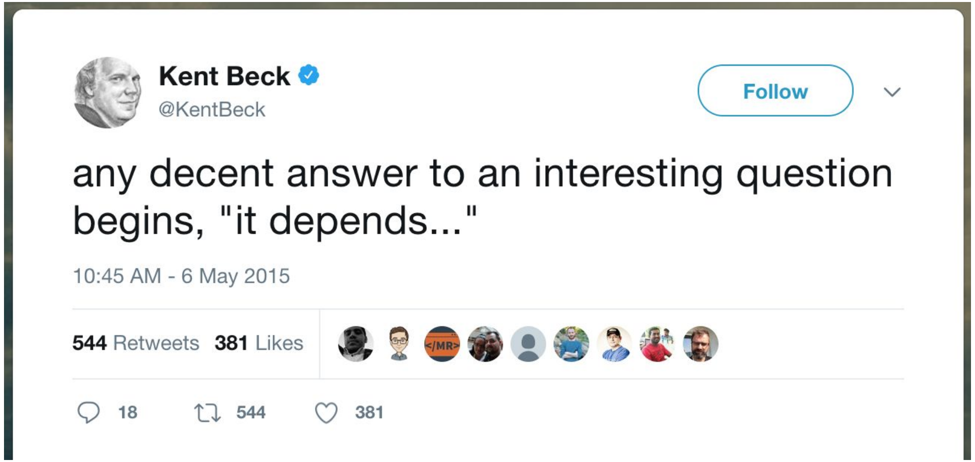
Many of the techniques we covered accelerate idea to prod

Theme I: Agile Triangle



We are always in the pursuit of value managing quality and constraints

Theme 2: My Final ‘It Depends’



Kent Beck
@KentBeck

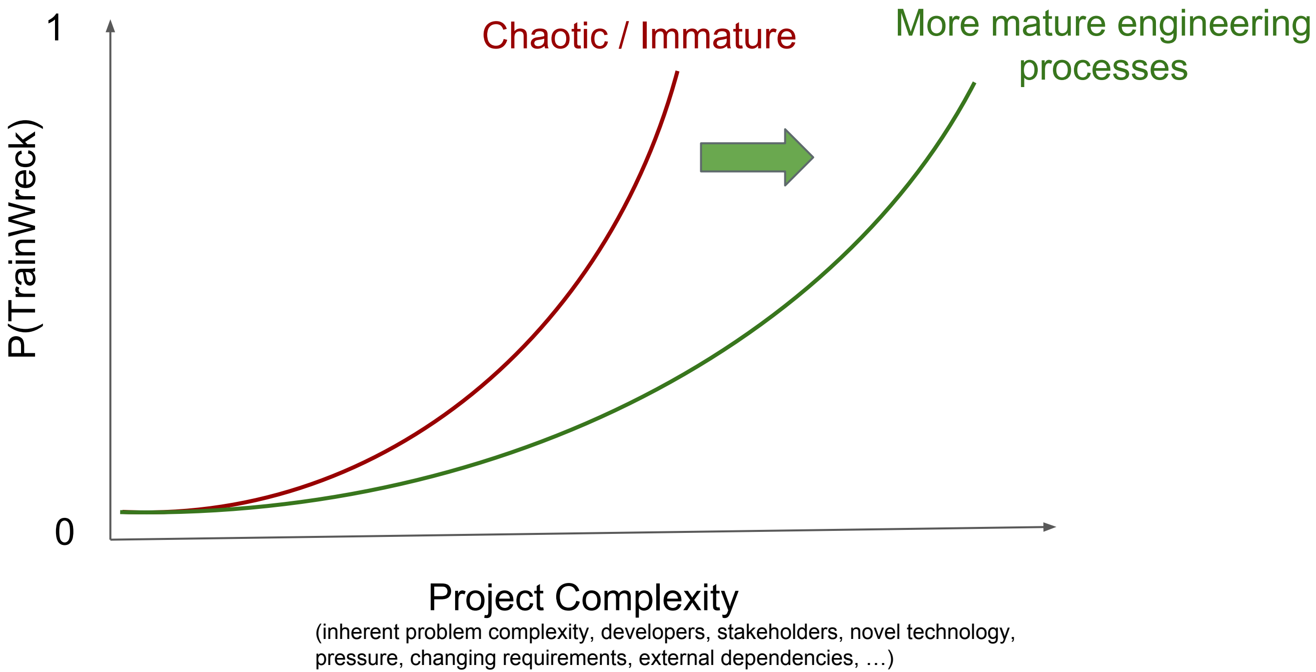
any decent answer to an interesting question begins, "it depends..."

10:45 AM - 6 May 2015

544 Retweets 381 Likes

But hopefully, we grew as engineers and have ***theoretical underpinnings*** and can ***evaluate situations***, understand ***trade-offs*** and ***pick optimal***

Theme 3: Complexity



Continual evolution to keep maturity > complexity

Theme 4: Awareness & Improvement

Continually reflect / retrospective your processes.
Diagnose and improve

W4156

Life Data Points

We are not hunting wooly mammoths any more

Was once told my impact function of:

- Problem Domain: do I understand what needs to be built?
- Engineering: can I build it?
- People skills: do I have the people skills to work with people to extract #1, deliver #2

Accelerate your experience through reading ....

> “A reader lives a thousand lives before he dies, said Jojen. The man who never reads lives only one.”
>
> — George R.R. Martin, A Dance with Dragons

- Of all the mistakes I have made most of them I have found the prophylaxis in a book at a later date
- Reading unsurprisingly accelerates your development
- Read vociferously: engineering, domain, data storytelling, leadership, etc

But have reasonable self-expectations

A lot of graduates place unreasonable expectations on themselves (there is a lot to know!).

However, the company ***chose*** to hire graduates. If they:

- want 10 years experience they would hire someone with >=10 years experience
- want 10 years experience from a graduate they are odd and you probably don't want to work for them anyway
- chose to hire graduates they want potential, motivation and some knowledge

At the graduate level it is one of the times you can say "I don't know but I want to/will find out". Do not set unrealistic/distorted expectations for yourself.

When faced with uncertainty ask “what data can I gather to reduce uncertainty”

(Intense stress) “Ewan, I don’t know what to do with my life/career! I don’t know which industry or role. I was thinking maybe Widgets and I like Product Management but I am not sure”

“Ok. (unfortunately - this is an ongoing issue. It applies equally at 30 as 20!). How can we help work it out?”

“I don’t know. I have ***luxuriated*** in my dorm room ***willing*** the answer from the skies but it has not arrived!”

“What about applying for a product manager role at a Widget company?”

“But Ewan, I said I am not *sure* if product management in widgets is for me. What if I don’t like the company or the role?”

“Fantastic! You have narrowed it down”

“Won’t I have wasted their time?”

“Interviews are a bilateral process/effort (they are working out if *you* are a fit for company and you are working out if *company* is fit for you).”

“Oh.”

“And equally, what if you find out some data point such as you *like* product mgt?”

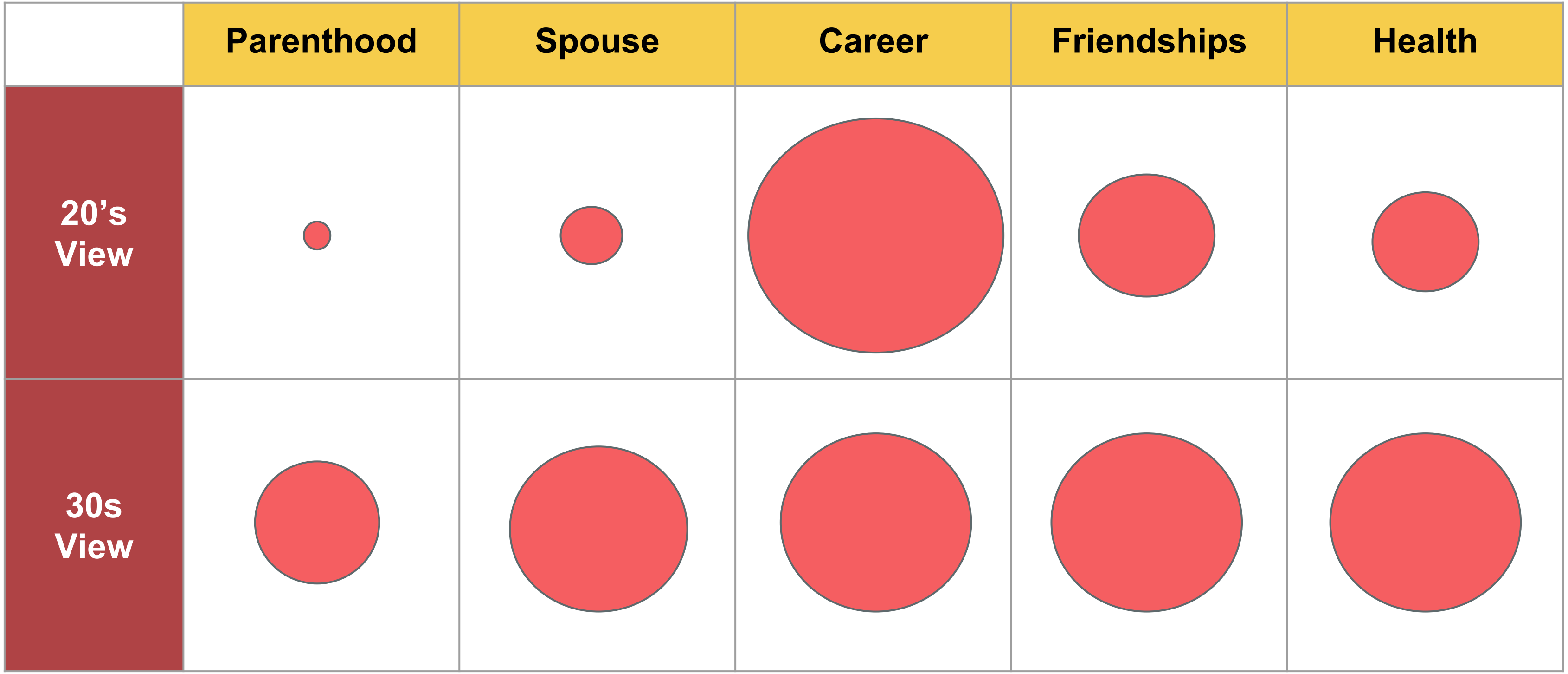
“Well, I suppose I know I confirmed product mgmt and now just need to find the company/industry”

“Absolutely. And vice versa (company, role)”

“So, I should treat job search as ‘search’ process/simulated annealing”

“Yes. Sometimes life is like simulated annealing”

But career isn’t everything … (The McConaughey 5)



	Parenthood	Spouse	Career	Friendships	Health
20’s View	●	●	●	●	●
30s View	●	●	●	●	●

Till I see you next

- Please stay in touch
  - Email
  - Or come back to say ‘hi’ in person
- Let me know how it goes at internships / industry
  - Did the course ease the transition?
- Happy to help as you find your way in industry
  Response rate
  - P(High): Problem X, Have considered A,B,C. Thinking D. Tradeoff E vs F. What do you think?
  - P(Low): Problem X. ?!?!