What makes a GREAT SOFTWARE ENGINEER?

PRESENTED BY:

Venus Pathak

Master in Computer Science

OUTLINE

- INTRODUCTION
- RELATED WORK
- QUESTIONS ADDRESSED
- METHODOLOGY
- FINDINGS
  - Personal Characteristics
  - Decision Making
  - Teammates
  - Software Product
- THREATS TO VALIDITY
- HOW DOES THIS PAPER HELP ?
- FUTURE WORK
- REFERENCES
- DISCUSSION

INTRODUCTION

0 DAYS

SINCE LAST NEW
JAVASCRIPT FRAMEWORK

INTRODUCTION

- According to ACM Computing Curricula, arguing that engineers need knowledge of technical areas and techniques such as programming fundamentals, verification/ validation, and project management.

NARROW DEFINITION

- people who produce software for earnest use

VAGUE

- Addresses the lack of specificity, breadth, and rigor

```
public  double abc (int a) {
double t;
double r = a / 2;
            do {
                        t = r;
            r = (t + (a / t)) / 2;
                        } while ((t - r) != 0);
 return r;
            }
```

```
// This has been done with reference to RR-77001(JIRA number)

// The following method calculates square root of given number

public double squareRoot(int number) {
double t;

double squareRoot = number / 2;

do {
      t = squareRoot;
      squareRoot = (t + (number / t)) / 2;
} while ((t - squareRoot) != 0);

return squareRoot;
}
```

INDENTATION

QUESTIONS ADDRESSED

- What do expert software engineers think are attributes of great software engineers?
- Why are these attributes important for the engineering of software?
- How do these attributes relate to each other?

RELATED WORK

- Why factors are (or are not) important ?
- Others considered related fields like Information Technologies / Information Systems
- Some factors/ attributes have been mentioned – key attributes not identified
- Narrow selection of factors

A COMPREHENSIVE SET OF ATTRIBUTES HAS BEEN IDENTIFIED AND DESCRIBED FOR THE FIRST TIME

METHODOLOGY

- 59 face-to-face semi-structured interviews
- 13 Microsoft divisions including several interviews with architect-level engineers with over 25 years of experience.

SELECTION CRITERIA FOR ENGINEERS

- At or above the Software Development Engineer Level 2 (SDEII) title
- At or above Senior Dev Manager - >15 years of experience
- confirmed as experts by other engineers

METHODOLOGY

- randomly sampled engineers in the 22 strata in a round-robin fashion
  - with 3 employees each round
  - at least 2 informants in each stratum

- Out of 152 - interviewed 59
- semi-structured interviews
- 1 hour in duration

METHODOLOGY

TABLE 1. STRATIFIED RANDOM SAMPLE OF EXPERIENCED ENGINEERS AT MICROSOFT

Experience Level \ Product Type	Ad Platform	Bing	Corp Dev	Dynamics	Office	Phone	Server & Tools	Windows	Windows Services	Xbox	Other	Totals
Experienced titles: SDE II, Senior SDE, Senior Dev Lead	2	2	2	2	2	3	3	6	3	2	2	29
Very Experienced titles: Architect, Technical Fellow, Partner Dev Manager, Partner Dev Lead, Principal Dev Lead, Senior Dev Manager, or Principal SDE	3	3	3	2	3	2	2	5	2	3	2	30
Totals	5	5	5	4	5	5	5	11	5	5	4	59


"Think back to someone you've worked with that you thought was a great software engineer. What were some attributes that made the person 'great' in your mind?"
Follow up and clarification questions for attributes that seemed interesting
we asked about attributes that either lacked clarity or might vary in interpretation
updated the set of attributes we inquired about (once every ~10 interviews)
limited discussions to 5 attributes of interest
Ground theory approach to analyze the more than 60 hours of interviews and 388,000 words of transcripts
Used open coding to identify and assess attributes of great software engineers
Selective coding pass through our data- consolidating the attribute set
help of a Senior Software Development Engineer (3rd author) to developing her own attributes
Consolidated with the initial set

FINDINGS

- Improving
- Passionate
- Open-minded
- Data-Driven

PERSONAL CHARACTERISTICS

- Knowledge about people and organization
- See the forest and the trees
- Update their mental models
- Handles complexity

DECISION MAKING

- Creates shared context
- Creates shared success
- Creates a safe haven
- Honest

TEAMMATES

- Elegant
- Creative
- Anticipates needs
- Carefully constructed

SOFTWARE PRODUCT

Personal Characteristics

- **IMPROVING : constantly looking to improve themselves**

"Computer technology, compared to other sciences or technology, it's pretty young. Every year there's some new technology, new ideas. If you are only satisfied with things you already learned, then you probably find out in a few years, you're out of date… good software engineer [sic], he keep investigate,
investment. [sic]" - Example: JavaScript frameworks

- **PASSIONATE : intrinsically interested in the area**

"I think that there are people who are great software engineers who are in the wrong place and aren't motivated and they end up not performing well."

- **OPEN – MINDED: willing to let new information change the way they think**

"You should be open… what you think need not be the right thing tomorrow… like the Facebook explosion, when Myspace was already there, but it exploded… no one knew that Facebook would explode when it started".

- **DATA – DRIVEN: decisions, when possible, should be made using data, not intuition or arguments**

"One thing that surprises me… even though we are driven by data, at least we try to believe we are… Some data gets shown to us. We figure out some ways to ignore it. So, maybe, maybe everybody thinks that they're data driven, but I've seen people come up with excuses for why the data doesn't apply to them. I've seen that a million times."

Decision Making

- **KNOWLEDGE ABOUT PEOPLE AND ORGANIZATION:** **informed about their coworkers' responsibilities, knowledge, and tendencies**

"Make sure that you are aware of that big picture, you know where you fit in and how you interact with everyone else to optimize what you are doing."

- **SEE THE FOREST AND THE TREES:** **considering situations at multiple levels of abstraction**

"What differentiated [this great engineer] from other people in management positions… capability to zoom into the details, and he was not just a high level guy, …know the reality of the stack or the reality of the software…" , **Example: scope**

- **UPDATE THEIR MENTAL MODELS:** **discarding old models for new ones, related to open-minded**

"You can always follow patterns too much… It's worked in the past, but conditions have changed. You always need to look and take a little bit of risk with each one of your tasks. If you're not then you're not really going to find out what's possible."

- **HANDLES COMPLEXITY:** **some software problems are inherently complex (handling with ease)**

"To solve the problem, [great engineers] have to have the ability to connect things… You are always debugging layers of stacks of code… this layer talks to some other layer in the horizontal… you need to solve the problem and you don't know what's going on."

Teammates

- **CREATES SHARED CONTEXT:** **communicating effectively**

"You perceive who you are talking to, and you are able to judge on those levels that they are, or you just ask important questions. Do you know about this? And then, be able to simplify the problem to the level that they're working in, or you estimate the amount of information given to them."

- **CREATES SHARED SUCCESS:** **success should be shared since goal is the same**

"No matter how good is our code, if our partner [sic] cannot give it a good product for us then we cannot share our greatness to the whole world. A lot of time I see our support to our client is not very well [sic]… we should have a good result combined together."

- **CREATES A SAFE HAVEN:** **engineers should not be afraid of making mistakes**

"I believe in having people feel the pain of their own mistakes… dealing with the ramifications of the decisions that are being made, I guess is the best way to learn."

- **HONEST:** **related to trust, accepting mistakes, not manipulating**

"Influence comes to someone else trusting you, part of that trust is that they go, 'You know what? I know that this person always speaks the truth.' As a result of that, when they say something is good, I will totally believe them because they are not trying to kind of misrepresent something or make them look better or whatever."

Software Product

- **ELEGANT:** creating simple designs and software, not complex

"The style… always, an idea, and it was all clean… very concise. Just looking at it, you can say, "Okay, this guy, he knew what he was doing."… There's no extra stuff. Everything is minimally necessary and sufficient as it should be. It's well thought-out off screen."

- **CREATIVE:** how creating you are depending on the existing solutions and their drawbacks

"If you're looking for really an innovative …or just a solution that's outside the current norm… think through the problem…constraints that are currently imposed on the environment."

- **ANTICIPATES NEEDS:** create software in a way that can accommodate future needs

"QQ, the Chinese chat program. It now has hundreds of millions of users. That system was designed fifteen years ago, when QQ only had a few million users. It still works today, that's amazing, to have a system that scales that well, to foresee all the issues it would have to face."

THREATS TO VALIDITY

- Only 1 hour for the interview
- Only 3 women among 59 informants
- Size of the organization may affect attributes related to people and organization
- May differ for other industries (non-software)

HOW DOES THIS PAPER HELP

- Software engineering is sociotechnical

CREATING A SAFE HAVEN **TEAMMATES** **CREATING SHARED SUCCESS**

- Effective decision making is important

- Ability to learn technical skill > an individual technical skill

HOW DOES THIS PAPER HELP

FOR RESEARCHERS

1. **New attributes:** Creating shared context, Creating a safe haven
2. **Motivation for a tool:** well-mannered in emails, evaluate trade-offs, see forest & trees, train engineers

FOR NOVICE ENGINEERS

1. **Attributes: list that they should aspire to achieve**
2. **Resumes, demonstrations**

FOR MANAGERS

1. **Improve:** help managers to become better
2. **Effective hiring decisions**
3. **Cultivate** attributes within his team

FOR EDUCATORS

1. **Curriculum choices, teaching methods and learning objectives**
2. **Examine teaching methods**

FUTURE WORK

- Study of combination of attributes
- Study of an individual attribute – empirical study
- Comparison of attributes in software engineering to attributes in other fields
- Relative importance of attributes
- Effect of factors like gender, background

REFERENCES

- What Makes A Great Software Engineer? By Paul Luo Li*+, Andrew J. Ko*, Jiamin Zhu+

DISCUSSION

- Can you think of any other attributes that could define a great software engineer ?
- Can these attributes change depending on the factors related to culture, organization or gender ?
- Do you think current education process inculcate all these factors? What can be done to improve it ?
- Which attributes do you think weigh more than the others ?

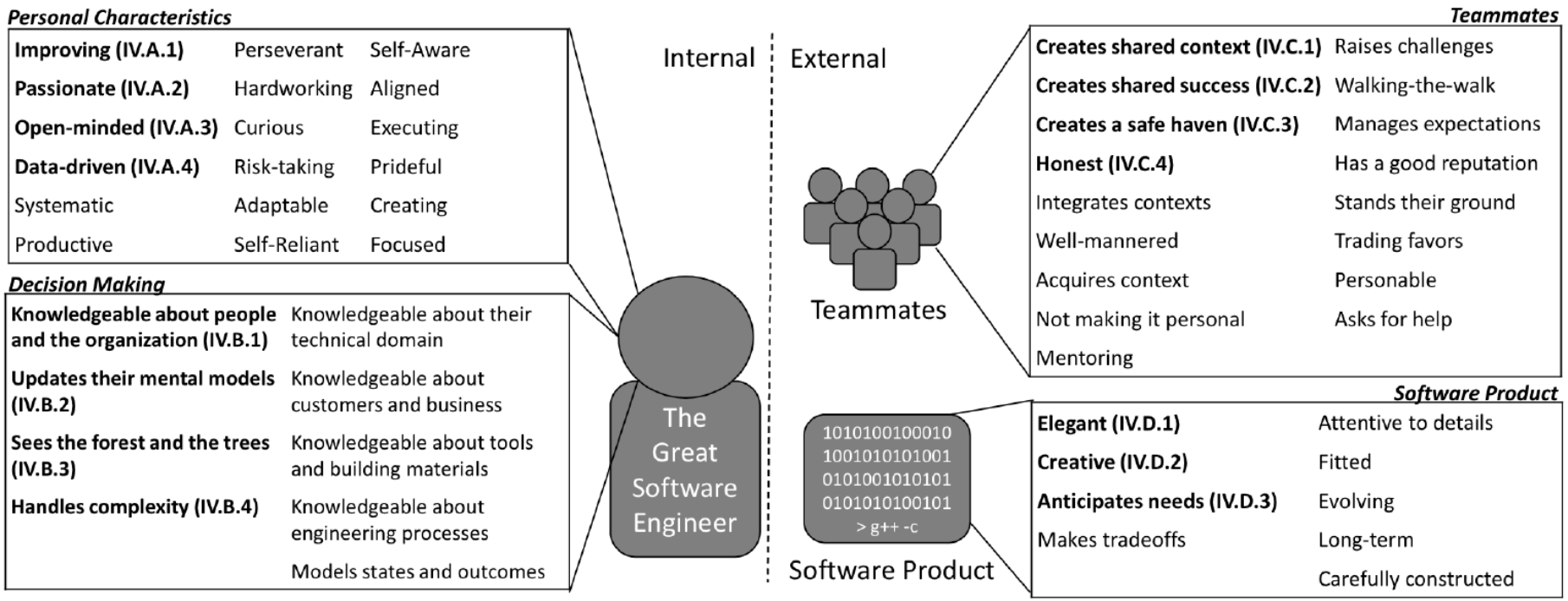

Fig. 1. Model of attributes of great software engineers, with attributes we discuss in detailed in bold.


?
THANK YOU
FOR
YOUR
ATTENTION
ANY QUESTIONS?