[CS3704] Software Engineering

Shawal Khalid
Virginia Tech
2/5/2024

Discussion Presentation

Example Presentation and Activity

Announcements

- **HW1 due tonight by 11:59pm**

What Makes A Great Software Engineer?

Authors: Paul Luo Li, Amy J. Ko, and Jiamin Zhu
Presenter: Shawal Khalid
International Conference on Software Engineering (ICSE) 2015

Problem

- Good software engineers are essential for developing high-quality software.
  - Companies want to hire them
  - Universities want to train them
  - Students/Novices want to be them
- **But, software engineers are difficult to evaluate!**
  - Technical skills, and beyond

Evaluation

- Interviewed 59 experienced software engineers at Microsoft.
  - Questions were mostly reflective
    - (i.e. *“Think back to someone you’ve worked with that you that was a great software engineer. What were some attributes that made the person ‘great’ in your mind?”*)
- Analyzed responses to derive 53 attributes of great software engineers.

Results

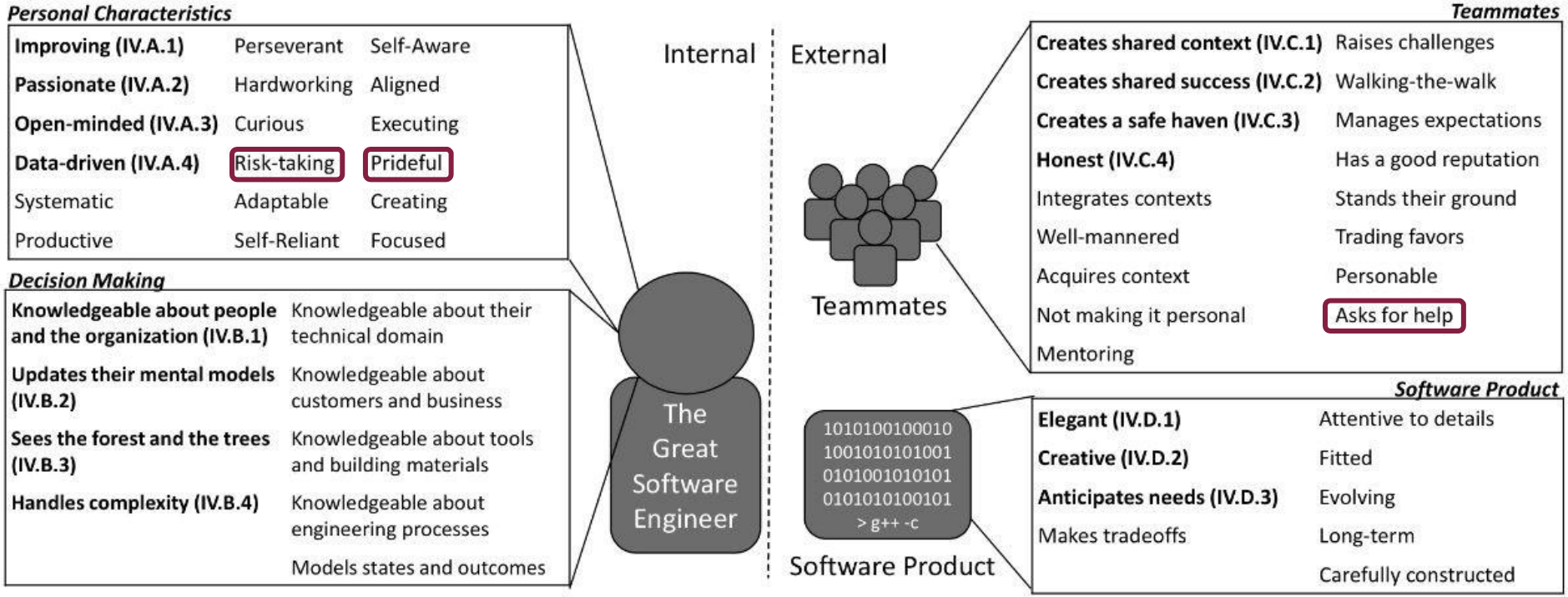

Fig. 1. Model of attributes of great software engineers, with attributes we discuss in detailed in bold.

Surprises

Risk-taking—willing to go into high-value areas even though they may not have knowledge or expertise (e.g. new technologies).

Prideful—taking pride in oneself and ones' product; letting their output be a reflection of their skills and trying their best to deliver.

Asks for help—finding and engaging others with needed knowledge and information.

Advantages of Research

- First study to analyze what makes a great software engineer.
- Authors provide implications for research, new software engineers, management, and education.

Advantages of Research

Researchers: Explore methods to measure *internal* attributes and develop tools and processes to increase *external* attributes.

Novice Software Engineers: Set of attributes to emulate, aspire to achieve, and to seek in potential mentors.

Managers: *"Walk the walk"*, create an environment and culture to foster these attributes with your development team.

Educators: Examine teaching methods and curricula, create a classroom environment and culture to foster these attributes.

Limitations

- Only conducted at Microsoft
  - *What about developers at other companies?*
- Does’t include quantitative data on results
  - i.e. *how many participants mentioned each attribute?*
- Lacks details on the background of participants
  - i.e. *average years of experience, demographic information, etc.*

Relevance to Class

Software engineering is a human activity, and software engineers use processes, methods, and tools to develop and maintain applications.

Today Was a Good Day: The Daily Life of Software Developers

Authors: Andre N. Meyer, Earl T. Barr, Christian Bird, and Thomas Zimmerman
Presenter: Shawal Khalid
IEEE Transactions on Software Engineering (TSE) 2021

Problem

- Software engineering is a complicated and chaotic process with many distractions.
- Good work days increase developer productivity, code quality, and job satisfaction.

Evaluation

Research Questions:

1. What factors influence good and typical developer workdays and how do they interrelate?
2. How do developers spend their time on a good and typical workday?*****
3. What are the different types of workdays and which ones are more often good and typical?*****
4. How does collaboration impact good and typical workdays?

Evaluation (cont.)

- Interviewed software engineers to discover work activities.
- Sent 37,792 surveys to developers and received 5,971 responses to characterize activities with workdays.
  - Types of workdays: Typical, Atypical, Good, and Bad

Results

TABLE 2
Mean and Relative Time Spent on Activities on Developers' Previous Workdays (WD)

Activity Category	All 100% (N=5928)		Typical WD 64% (N=3750)		Atypical WD 36% (N=2099)		Good WD 61% (N=3028)		Bad WD 39% (N=1970)	
	pct	min	pct	min	pct	min	pct	min	pct	min
Development-Heavy Activities										
Coding (reading or writing code and tests)	15%	84	17%	92	13%	70	18%	96	11%	66
Bugfixing (debugging or fixing bugs)	14%	74	14%	77	12%	68	14%	75	13%	72
Testing (running tests, performance/smoke testing)	8%	41	8%	44	7%	36	8%	43	7%	38
Specification (working on/with requirements)	4%	20	3%	17	4%	25	4%	20	4%	20
Reviewing code	5%	25	5%	26	4%	23	4%	24	5%	26
Documentation	2%	9	1%	8	2%	10	2%	9	2%	8
Collaboration-Heavy Activities										
Meetings (planned and unplanned)	15%	85	15%	82	17%	90	14%	79	18%	95
Email	10%	53	10%	54	10%	54	9%	52	10%	57
Interruptions (impromptu sync-up meetings)	4%	24	4%	25	4%	22	4%	22	5%	28
Helping (helping, managing or mentoring people)	5%	26	5%	27	5%	25	5%	26	5%	28
Networking (maintaining relationships)	2%	10	2%	9	2%	12	2%	11	2%	10
Other Activities										
Learning (honing skills, continuous learning, trainings)	3%	17	3%	14	4%	22	3%	19	3%	16
Administrative tasks	2%	12	2%	11	3%	14	2%	11	3%	15
Breaks (bio break, lunch break)	8%	44	8%	44	8%	45	8%	44	8%	45
Various (*e.g.*, traveling, planning, infrastructure set-up)	3%	21	3%	17	5%	27	3%	19	4%	25
Total	**9.08** hours		**9.12** hours		**9.05** hours		**9.17** hours		**9.15** hours	

The left number in a cell indicates the average relative time spent (in percent) and the right number in a cell the absolute average time spent (in minutes).

Unsurprising

TABLE 2
Mean and Relative Time Spent on Activities on Developers' Previous Workdays (WD)

Activity Category	All 100% (N=5928)		Typical WD 64% (N=3750)		Atypical WD 36% (N=2099)		Good WD 61% (N=3028)		Bad WD 39% (N=1970)	
	pct	min	pct	min	pct	min	pct	min	pct	min
Development-Heavy Activities										
Coding (reading or writing code and tests)	15%	84	17%	92	13%	70	18%	96	11%	66
Bugfixing (debugging or fixing bugs)	14%	74	14%	77	12%	68	14%	75	13%	72
Testing (running tests, performance/smoke testing)	8%	41	8%	44	7%	36	8%	43	7%	38
Specification (working on/with requirements)	4%	20	3%	17	4%	25	4%	20	4%	20
Reviewing code	5%	25	5%	26	4%	23	4%	24	5%	26
Documentation	2%	9	1%	8	2%	10	2%	9	2%	8
Collaboration-Heavy Activities										
Meetings (planned and unplanned)	15%	85	15%	82	17%	90	14%	79	18%	95
Email	10%	53	10%	54	10%	54	9%	52	10%	57
Interruptions (impromptu sync-up meetings)	4%	24	4%	25	4%	22	4%	22	5%	28
Helping (helping, managing or mentoring people)	5%	26	5%	27	5%	25	5%	26	5%	28
Networking (maintaining relationships)	2%	10	2%	9	2%	12	2%	11	2%	10
Other Activities										
Learning (honing skills, continuous learning, trainings)	3%	17	3%	14	4%	22	3%	19	3%	16
Administrative tasks	2%	12	2%	11	3%	14	2%	11	3%	15
Breaks (bio break, lunch break)	8%	44	8%	44	8%	45	8%	44	8%	45
Various (*e.g.*, traveling, planning, infrastructure set-up)	3%	21	3%	17	5%	27	3%	19	4%	25
Total	**9.08** hours		**9.12** hours		**9.05** hours		**9.17** hours		**9.15** hours	

The left number in a cell indicates the average relative time spent (in percent) and the right number in a cell the absolute average time spent (in minutes).

Surprises

TABLE 2
Mean and Relative Time Spent on Activities on Developers' Previous Workdays (WD)

Activity Category	All 100% (N=5928)		Typical WD 64% (N=3750)		Atypical WD 36% (N=2099)		Good WD 61% (N=3028)		Bad WD 39% (N=1970)	
	pct	min	pct	min	pct	min	pct	min	pct	min
Development-Heavy Activities										
Coding (reading or writing code and tests)	15%	84	17%	92	13%	70	18%	96	11%	66
Bugfixing (debugging or fixing bugs)	14%	74	14%	77	12%	68	14%	75	13%	72
Testing (running tests, performance/smoke testing)	8%	41	8%	44	7%	36	8%	43	7%	38
Specification (working on/with requirements)	4%	20	3%	17	4%	25	4%	20	4%	20
Reviewing code	5%	25	5%	26	4%	23	4%	24	5%	26
Documentation	2%	9	1%	8	2%	10	2%	9	2%	8
Collaboration-Heavy Activities										
Meetings (planned and unplanned)	15%	85	15%	82	17%	90	14%	79	18%	95
Email	10%	53	10%	54	10%	54	9%	52	10%	57
Interruptions (impromptu sync-up meetings)	4%	24	4%	25	4%	22	4%	22	5%	28
Helping (helping, managing or mentoring people)	5%	26	5%	27	5%	25	5%	26	5%	28
Networking (maintaining relationships)	2%	10	2%	9	2%	12	2%	11	2%	10
Other Activities										
Learning (honing skills, continuous learning, trainings)	3%	17	3%	14	4%	22	3%	19	3%	16
Administrative tasks	2%	12	2%	11	3%	14	2%	11	3%	15
Breaks (bio break, lunch break)	8%	44	8%	44	8%	45	8%	44	8%	45
Various (*e.g.*, traveling, planning, infrastructure set-up)	3%	21	3%	17	5%	27	3%	19	4%	25
Total	**9.08** hours		**9.12** hours		**9.05** hours		**9.17** hours		**9.15** hours	

The left number in a cell indicates the average relative time spent (in percent) and the right number in a cell the absolute average time spent (in minutes).

Advantages of Research

- Huge sample size (5,000+ responses)
- Thorough statistical analysis on data collected through survey
- Implications for optimizing developer workdays, evaluating success, and measuring productivity in SE work.
  - *How do we make good days typical?*
  - *Productivity not just defined by code!*

Limitations

- Only conducted at Microsoft
  - *What about developers at other companies?*
- “Goodness” and “badness” are relative
- The typicality and goodness of workdays are not binary
- Difficult for software engineer to self-report how much of your day was spent doing specific tasks

Relevance to class

- **Relevance to this class:** Software engineering tasks and artifacts (work products) correlate with good and bad days for developers.
  - Ex) On average, good workdays have less documentation, requirements analysis, etc.

Activity: 1-2-4-All

- Take a few minutes to individually think about the discussion topic. [1]
  - Write down your thoughts and ideas.
- Find a partner in class to discuss your thoughts with, come up with four main points. [2]
- Share with the class [All]

Activity: 1-2-4-All

*How can SE courses (like this one) be better designed and structured to help students become **great software engineers** and have typically **good workdays** in their careers?*

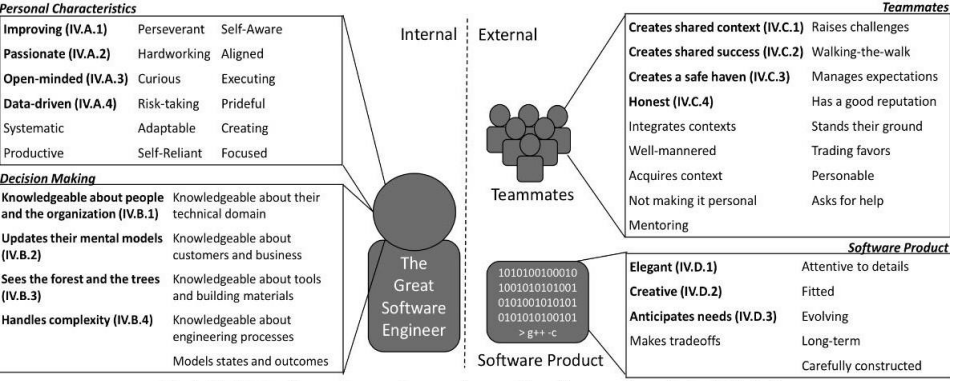

Fig. 1. Model of attributes of great software engineers, with attributes we discuss in detailed in bold.

Activity Category	All 100% (N=5928) pct	All min	Typical WD 64% (N=3750) pct	Typical WD min	Atypical WD 36% (N=2099) pct	Atypical WD min	Good WD 61% (N=3028) pct	Good WD min	Bad WD 39% (N=1970) pct	Bad WD min
Development-Heavy Activities										
Coding (reading or writing code and tests)	15%	84	17%	92	13%	70	18%	96	11%	66
Bugfixing (debugging or fixing bugs)	14%	74	14%	77	12%	68	14%	75	13%	72
Testing (running tests, performance/smoke testing)	8%	41	8%	44	7%	36	8%	43	7%	38
Specification (working on/with requirements)	4%	20	3%	17	4%	25	4%	20	4%	20
Reviewing code	5%	25	5%	26	4%	23	4%	24	5%	26
Documentation	2%	9	1%	8	2%	10	2%	9	2%	8
Collaboration-Heavy Activities										
Meetings (planned and unplanned)	15%	85	15%	82	17%	90	14%	79	18%	95
Email	10%	53	10%	54	10%	54	9%	52	10%	57
Interruptions (impromptu sync-up meetings)	4%	24	4%	25	4%	22	4%	22	5%	28
Helping (helping, managing or mentoring people)	5%	26	5%	27	5%	25	5%	26	5%	28
Networking (maintaining relationships)	2%	10	2%	9	2%	12	2%	11	2%	10
Other Activities										
Learning (honing skills, continuous learning, trainings)	3%	17	3%	14	4%	22	3%	19	3%	16
Administrative tasks	2%	12	2%	11	3%	14	2%	11	3%	15
Breaks (bio break, lunch break)	8%	44	8%	44	8%	45	8%	44	8%	45
Various (*e.g.*, traveling, planning, infrastructure set-up)	3%	21	3%	17	5%	27	3%	19	4%	25
Total	**9.08** hours		**9.12** hours		**9.05** hours		**9.17** hours		**9.15** hours	

The left number in a cell indicates the average relative time spent (in percent) and the right number in a cell the absolute average time spent (in minutes).

Next Class…

- **No class Monday (2/12)**
  - **Go to the career fair**
- **HW1 (due tonight at 11:59pm)**

- **Requirements Analysis next class**