Requirements Engineering

(Summer 2022)

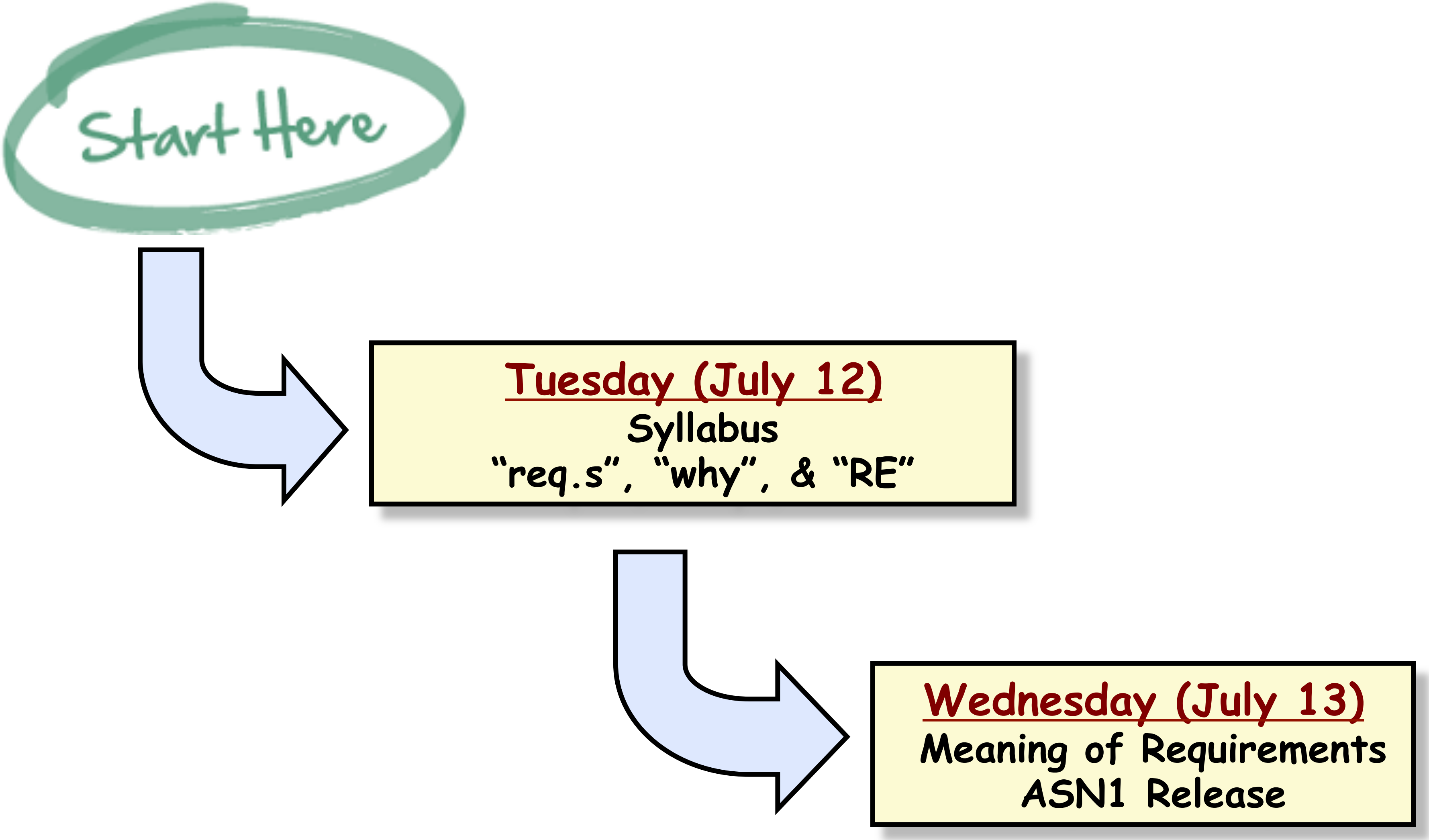
Prof. Nan Niu (nan.niu@uc.edu)

https://github.com/nanniu/RE-Summer2022

Please use your real name for Zoom

- For example:
  - Nan Niu
  - Jinzhi Shan
- Zoom poll to break the ice

Today's Menu



Tuesday (July 12)
Syllabus
"req.s", "why", & "RE"

Wednesday (July 13)
Meaning of Requirements
ASN1 Release

Today's take-aways

→ What're requirements?

→ Why're requirements important?

→ What's requirements engineering (RE)?
 - ↳ *Will sort out on July 13*

About This Course: Logistics

- Three lectures on each weekday
  - 9:00am-9:40am
  - 9:50am-10:30am
  - 10:40am-11:20am
- To make it appealing
  - Take-away of the day
  - Side note of the day
  - Class participation from you
- All slides & videos will be shared *after* each day's lectures
  - I need you to stay focused
  - Yet, you don't have to copy (or screenshot) crazily
- Communications shall be done via
  - WeChat group (if the issue is applicable to the entire class)
  - Email (nan.niu@uc.edu) [if the issue is private to yourself]

About This Course: Past & Now

→ Summer 2018	in person
→ Summer 2019	in person
→ Summer 2021	online
→ Summer 2022	online

About The Instructor

- Where's he coming from?
  - Hometown: Beijing, China
  - B.Eng. (1995-1999), Beijing Institute of Technology, China
  - Programmer (1999-2001), Lenovo, China
  - M.Sc. (2001-2004), University of Alberta, Canada
  - Ph.D. (2004-2009), University of Toronto, Canada
  - Assistant Professor (2009-2014), Mississippi State Univ.
  - Assistant Professor (2014-2018), Univ. of Cincinnati, USA
  - Associate Professor (2018-Present), Univ. of Cincinnati
- What's his research area?
  - Software Engineering, RE, Scientific Software Development, Human-Centric Computing
- How to obtain course-related information?

  https://github.com/nanniu/RE-Summer2022 *(will be updated)*
- Anything else?

About Me

↳Associate Editor

Springer

↳Steering Committee Member

ICSR
THE 20TH INTERNATIONAL CONFERENCE
ON SOFTWARE & SYSTEMS REUSE
MONTPELLIER FRANCE 2022 – 15/16/17 JUNE

↳Organizer

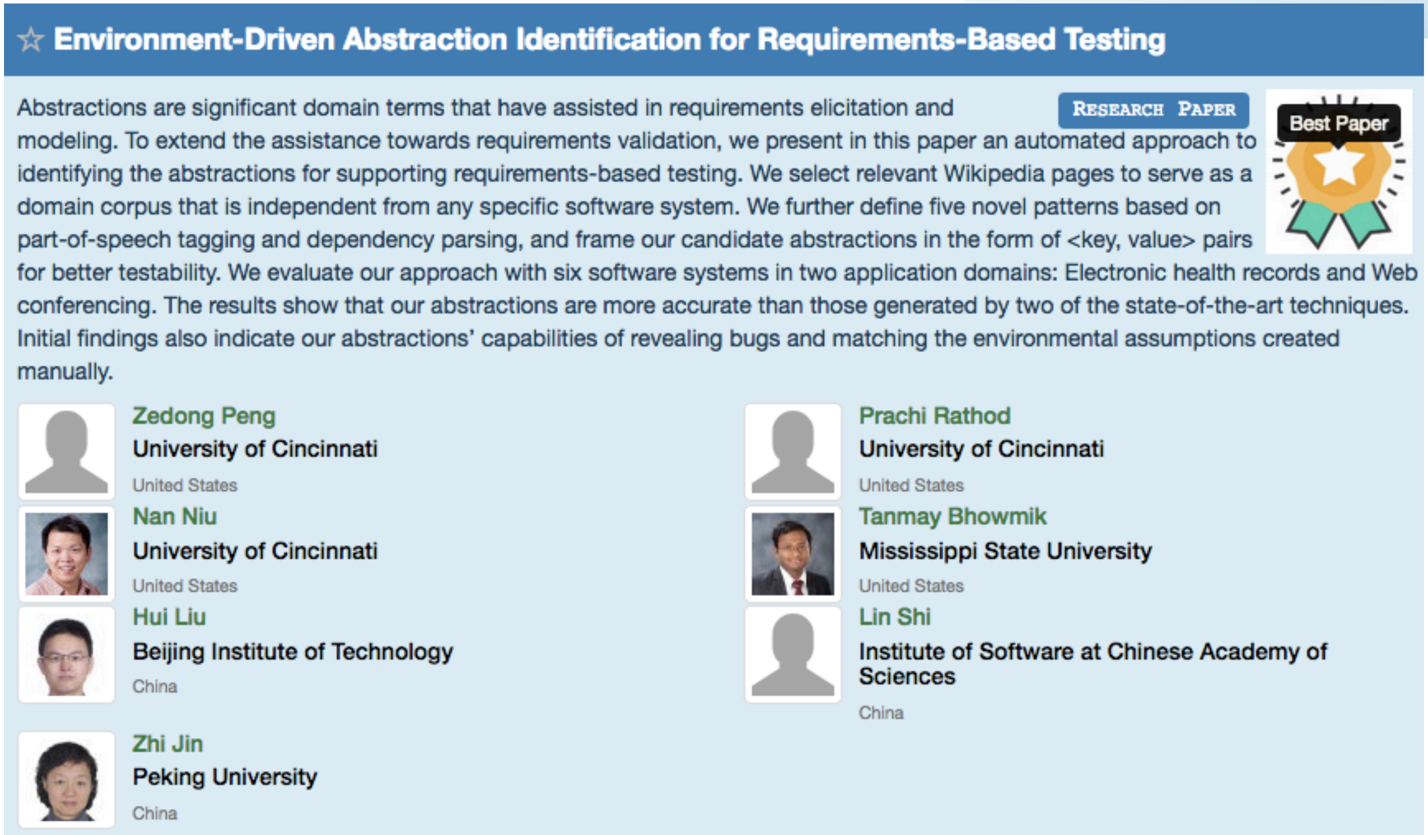
Disclaimer

→ All the pictures used in this course are from Google Images (https://images.google.com/), unless otherwise specified.

My Recent Papers

Deep Learning Based Program Generation from Requirements Text: Are We There Yet?

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING



☆ Environment-Driven Abstraction Identification for Requirements-Based Testing

RESEARCH PAPER

Best Paper

Abstractions are significant domain terms that have assisted in requirements elicitation and modeling. To extend the assistance towards requirements validation, we present in this paper an automated approach to identifying the abstractions for supporting requirements-based testing. We select relevant Wikipedia pages to serve as a domain corpus that is independent from any specific software system. We further define five novel patterns based on part-of-speech tagging and dependency parsing, and frame our candidate abstractions in the form of <key, value> pairs for better testability. We evaluate our approach with six software systems in two application domains: Electronic health records and Web conferencing. The results show that our abstractions are more accurate than those generated by two of the state-of-the-art techniques. Initial findings also indicate our abstractions' capabilities of revealing bugs and matching the environmental assumptions created manually.

Zedong Peng
University of Cincinnati
United States

Prachi Rathod
University of Cincinnati
United States

Nan Niu
University of Cincinnati
United States

Tanmay Bhowmik
Mississippi State University
United States

Hui Liu
Beijing Institute of Technology
China

Lin Shi
Institute of Software at Chinese Academy of Sciences
China

Zhi Jin
Peking University
China

About This Course: Objectives

- → Examine the state-of-the-art of research & current practice in RE
  - ➾ Role of RE in software and systems engineering
  - ➾ Current techniques, notations, methods, processes, and tools used in RE
- → Gain practical experience in selected RE techniques
- → Understand the essential nature of RE
  - ➾ Breadth of skills needed for RE, and the many disciplines on which it draws
  - ➾ Contextual factors & practicalities
- → Gain a basic grounding for research in RE
  - ➾ Methodological issues for RE research
  - ➾ Current research issues & direction of the field
  - ➾ Awareness of the literature

Course Assessments

→ Individual assignments

- 4 (modeling & programming) x 20 points = 80 points
- 1 (oral presentation) x 10 points = 10 points
- Class participation (poll, whiteboard, break out, etc.): 10 points
- Total: 100 points

Class Participation #2

- Go into the breakout room with your classmates
- Introduce yourself to each other
- Together as a group, define “requirements” in *one sentence*
  - No web search, please
- When your group is done, leave the breakout room and/or send a chat to me
- When every group is done, you’ll share your group’s answer on the Zoom whiteboard

What're "Requirements"?

- → Requirements = stakeholders' needs and desires

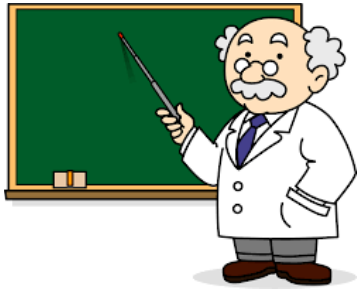
- → Stakeholders = those who have a stake in the change being considered & who stand to gain or lose from the change

  - ➪ The holders of the bets in a gambling game

  - ➪ Discussion: Who're the <u>STAKEHOLDERS</u> of Zoom (<u>https://ucincinnati.zoom.us/j/4879043098</u>) and what're their requirements (needs and desires)?

Stakeholders of Zoom

https://ucincinnati.zoom.us/j/4879043098



Builder: software developers, testers, marketing & sales …

User: customers (those who pay), users (those who use) …

Contexts: social, legal, ethical, technological, organizational …

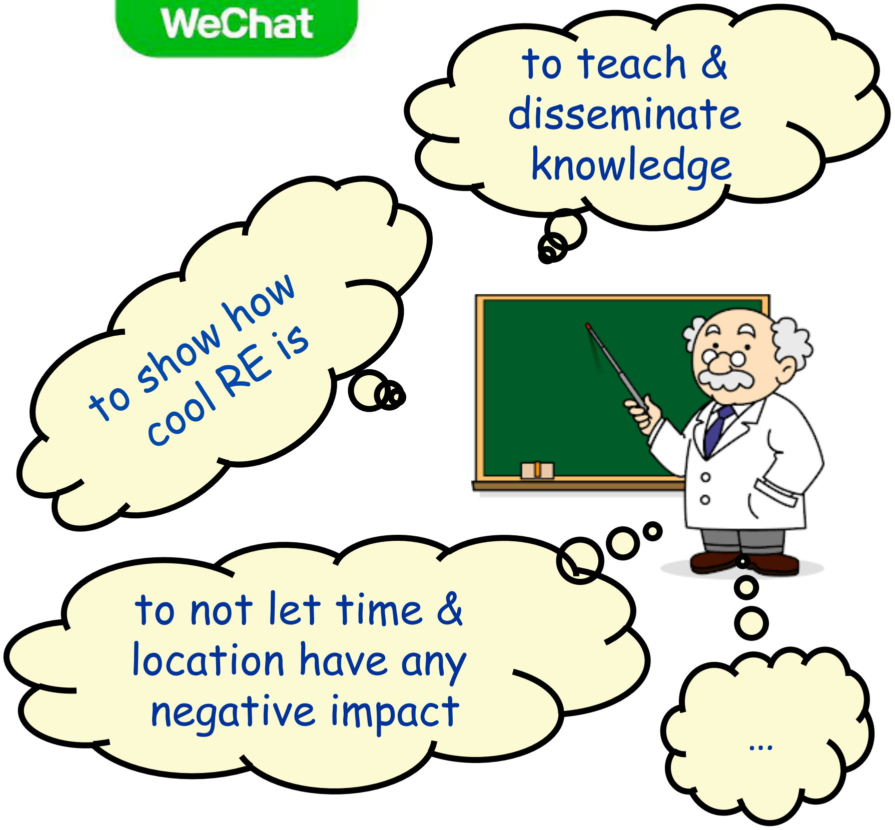
Stakeholders' needs & desires

GitHub

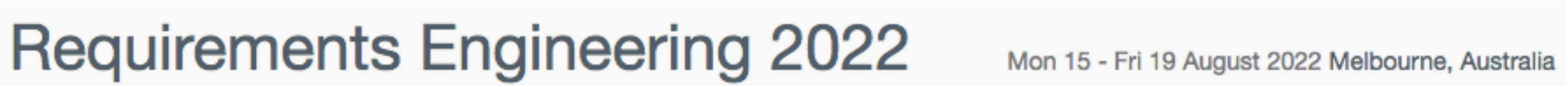
zoom

WeChat

Google Drive


to teach & disseminate knowledge
to show how cool RE is
to not let time & location have any negative impact
…

Stakeholders' needs & desires



Requirements Engineering 2022 Mon 15 - Fri 19 August 2022 Melbourne, Australia

→ Always an important RE topic & will continue to be an important RE topic in the days of AI

https://conf.researchr.org/track/RE-2022/RE-2022-Research-Papers#Call-for-Papers

Today's take-aways

- → What're requirements?
  - ↳ Stakeholders' needs & desires
- → Why're requirements important?
- → What's requirements engineering (RE)?

Your conclusion: ________________

- "48% of the failures observed in a medium-scaled software project were attributed to incorrect or misinterpreted functional specifications or requirements"

  V. Basili and B. Perricone,"Software errors and complexity: an empirical investigation", *CACM'84*

- "79.6% of interface faults were due to incomplete or omitted requirements"

  D. Perry and C. Stieg,"Software faults in evolving a large, real-time system: a case study", *ESEC'93*

- NASA spacecraft software systems: "the primary cause of safety-related functional faults is errors in recognizing (understanding) the requirements (62% on Voyager; 79% on Galileo)"

  R. Lutz, "Analyzing software requirements errors in safety-critical, embedded systems", *RE'93*

Requirements faults are common.

Your conclusion: ________________

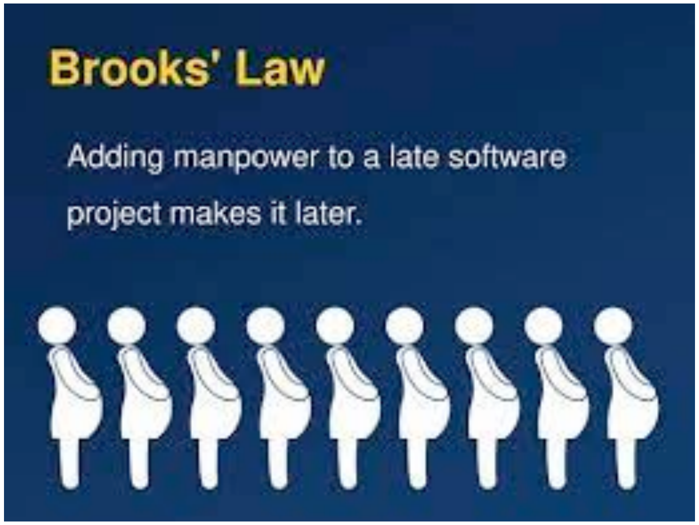
"The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is as difficult to rectify later."

Frederick P. Brooks, Jr.

"Clearly, it pays off to invest effort in finding requirements errors early and correcting them, say 1 man-hour rather than waiting to find the error during operations and having to spend 100 man-hours correcting it."

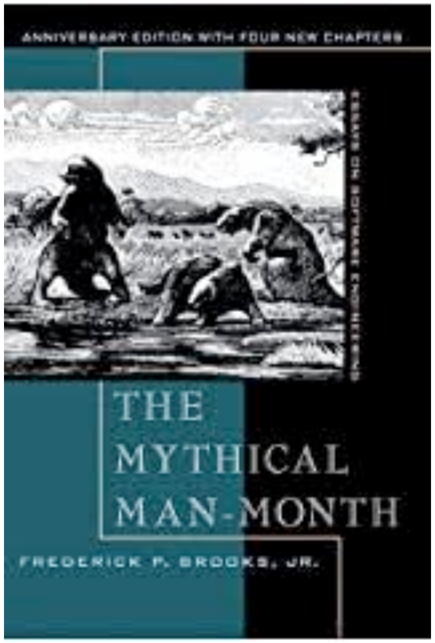
Barry Boehm

Side Note



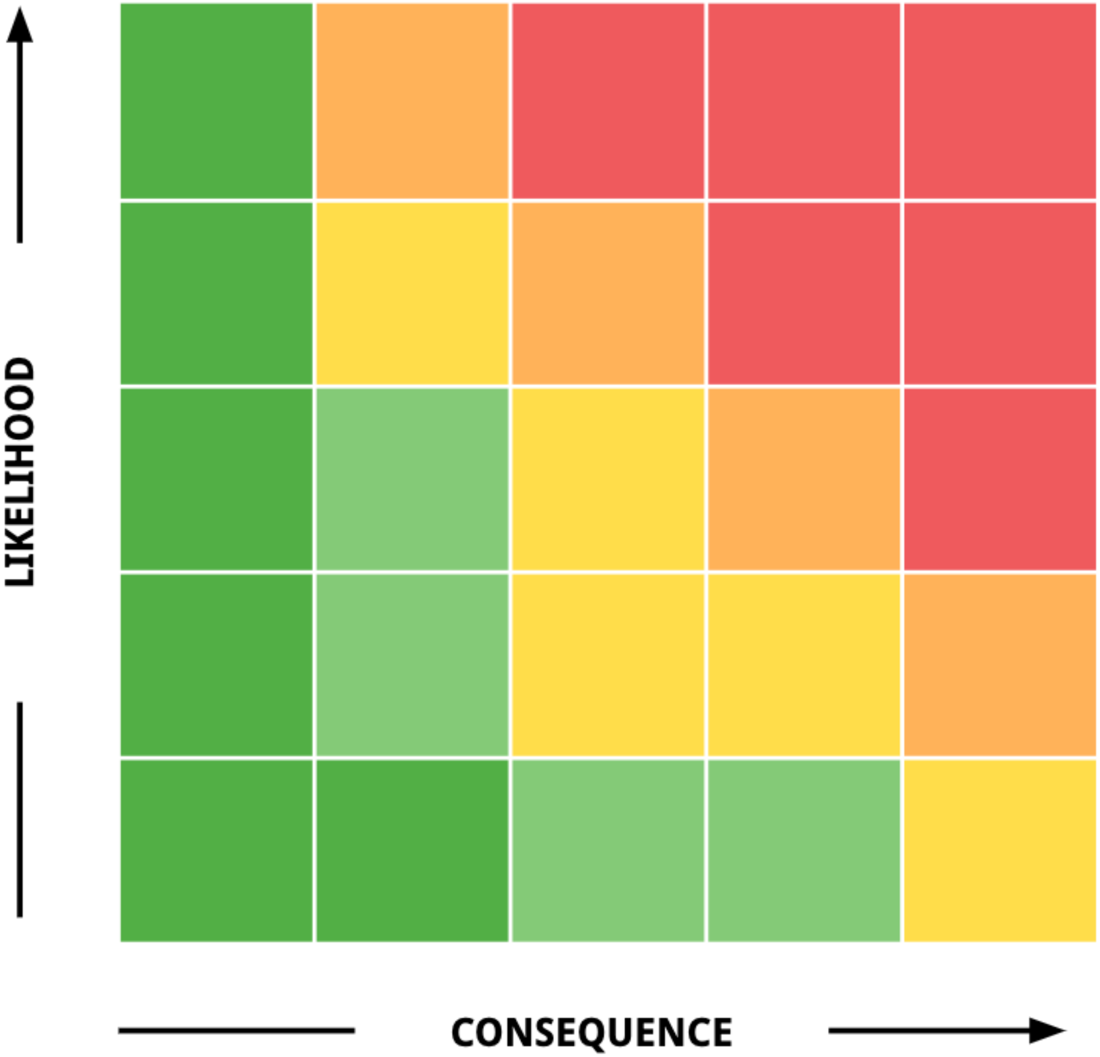
Brooks' Law

Adding manpower to a late software project makes it later.


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
ESSAYS ON SOFTWARE ENGINEERING
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.

Requirements faults are costly.

NASA's Risk Matrix



What's the problem?

A baseball team's GM & scouts

BINGE
SOCIETY
0:27 / 3:16
© 2011 Columbia Pictures Industries, Inc. All Rights Reserved.

Okay, good.
What's the problem?

Source: https://www.youtube.com/watch?v=viUiDANtRJc

2002 MLB Team Salaries

$125 million

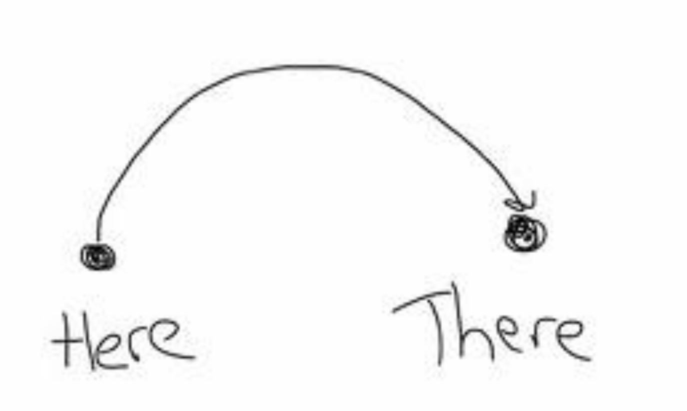
(highest)

$44 million

(3rd lowest)

Source: *https://en.wikipedia.org/wiki/Moneyball*

Problem matters



A problem is a *difference*

between things as desired
and things as perceived

Scouts: replacing expensive players with less expensive ones

© 2022, Nan Niu

Source: https://www.youtube.com/watch?v=TPNfhxbDLuw

Problem matters *a lot*

- Scouts
  - Replacing expensive players with less expensive ones
- Newly recognized needs & desires
  - Goal should *NOT* be to buy players
  - Goal should be to buy wins
    - In order to buy wins, you need to buy runs
- … and yes, the problem (READ: requirements) changes everything

Recap so far

- Requirements are important *because*
  - It's hard (*hardest*) to get them right.
  - It's common to get them wrong.
  - Getting them wrong is costly.
    - Some ($) numbers on the next slide
  - Hence, *"doing requirements right saves money"*.

Importance of RE (some numbers)

- Problems
  - Increased reliance on software
    - E.g. cars, dishwashers, cell phones, web services, …
    - Philips estimates that the amount of software in consumer products is doubling every two years
  - Software now the biggest cost element for mission critical systems
    - E.g. Boeing 777: approximately 50% of entire development cost for the plane was spent on the avionics software
  - Wastage on failed projects
    - E.g. 1997 GAO (US Government Accountability Office) report: $145 billion over 6 years on software that was never delivered
  - High consequences of failure
    - E.g. Ariane 5: $500 million payload
    - E.g. Intel Pentium bug: $475 million
- Key factors:
  - Certification costs
    - E.g. Boeing 777: >40% of software budget spent on testing to FAA standards compliance
  - Re-work from defect removal
    - E.g. Motorola: 60-80% of software budget (was) spent on re-work
  - Changing Requirements
    - E.g. California DMV (Dept of Motor Vehicles) system was cancelled after 6 years, after spending $44 million
    - E.g. London ambulance dispatch system; Denver airport luggage management system

Importance of RE (most memorable #)

Software defects cost 50-200 times more to correct once fielded than they would if they were detected during requirements engineering.

Source: *S. McConnell, "From the editor – an ounce of prevention", IEEE SW., 18(3):5-7, 2001.*

Try to read Jackson's paper

- A *highly recommended* reading **BEFORE** tomorrow's class (July 13), "Meaning of Requirements", can be found in the course website (in the "Readings" folder)