



# Informatics 43

## LECTURE 2

“WHAT IS SOFTWARE ENGINEERING? (PART 2)”

# Last Lecture



- Software is everywhere
  - Cars, medical field, politics, military, space, disaster response, games, law enforcement, ...
- Software is...
  - exciting, fun, versatile, helpful, harmful
  - large, built by teams, long-lived, multi-version, complex
- Informatics 43 definition of software engineering
  - The process of constructing software
  - Phases of development other than programming
  - Principles and qualities of enduring value

# Today's Lecture – What is software engineering?

- Perspectives on software engineering
- “Essential ingredients”
- Software Principles
- No Silver Bullet
- Quiz 1 study guide

# Today's Lecture – What is software engineering?

- Perspectives on software engineering
- “Essential ingredients”
- Software Principles
- No Silver Bullet
- Quiz 1 study guide



***Shawna,  
CEO***



***Andy,  
Software Engineer***



***Clara,  
Working mother***



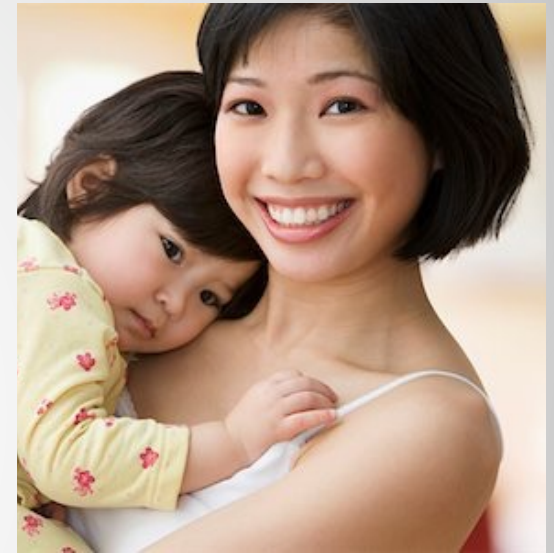
# “What is software engineering?”



***Shawna,  
CEO***



***Andy,  
Software Engineer***



***Clara,  
Working mother***

# Shawna's Perspective



- Cars: "...of all the **staff hours** in the entire program to build the Two-Mode Hybrid transmission...some 70 percent...were devoted to developing the control software"
- Medical systems: "...they won't cover anywhere near the **staggering cost** of an Epic EHR. Duke University Health System will shell out **\$700 million**; University of California will pay **\$150 million**..."
- Video games: "The video game industry is richer than it has ever been. Its revenue in 2018 was **\$43.8 billion**, a recent report estimated, thanks in large part to hugely popular games like Fortnite and Call of Duty."

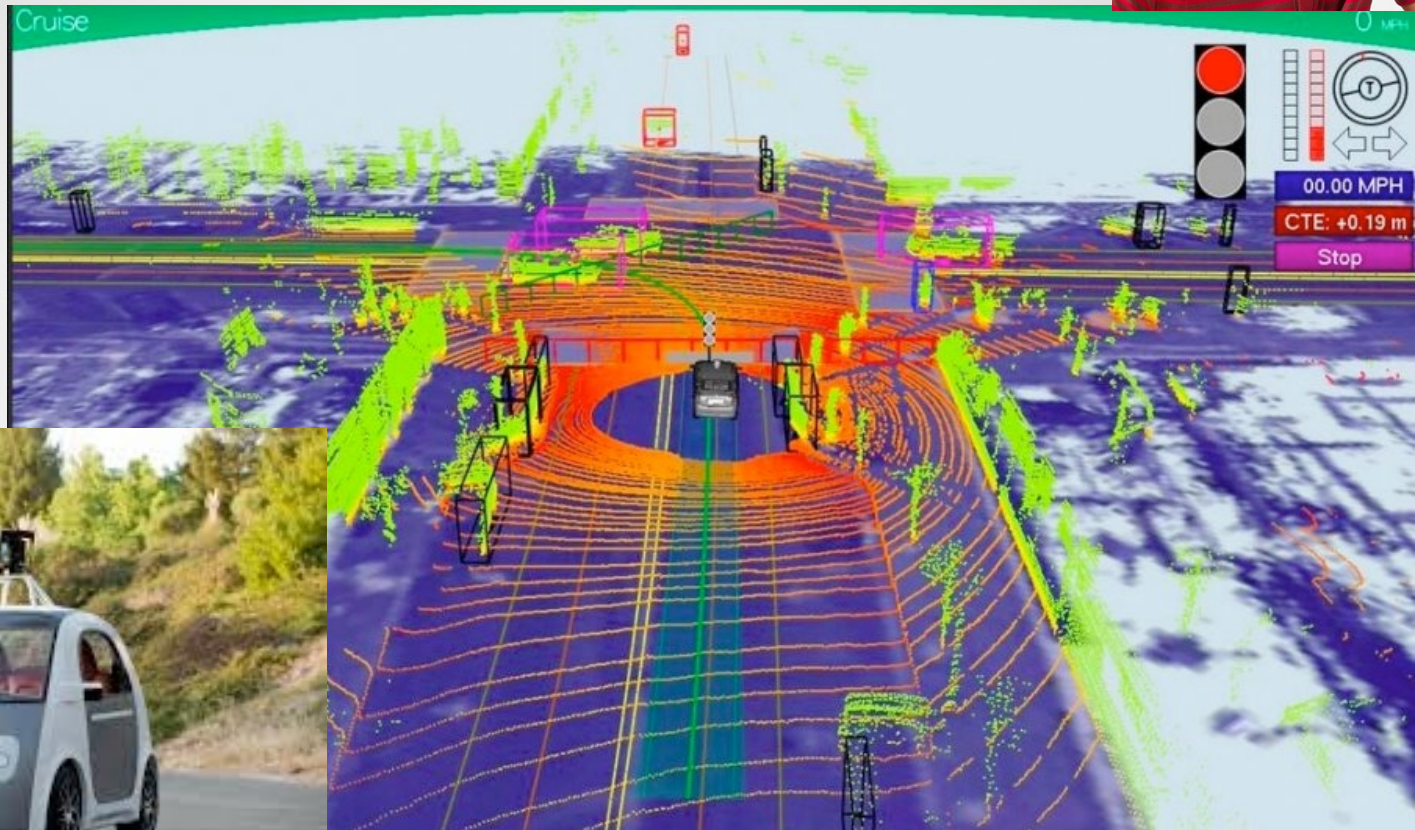
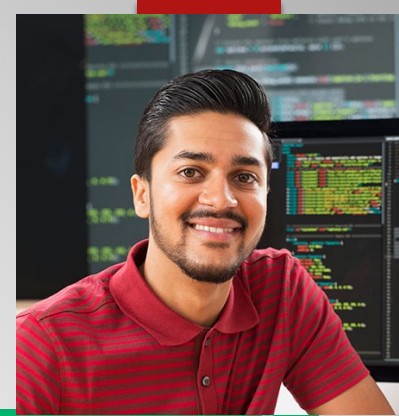
# Shawna = **Business** perspective

- What matters most?
- How does software engineering help?





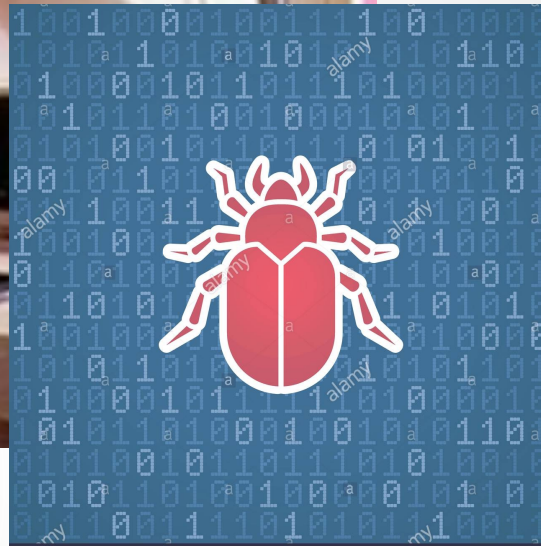
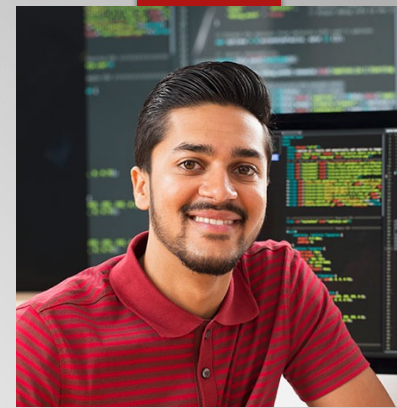
# Andy's Perspective: Cars



<https://www.youtube.com/watch?v=tiwVMrTLUWg> (start at 7:44)

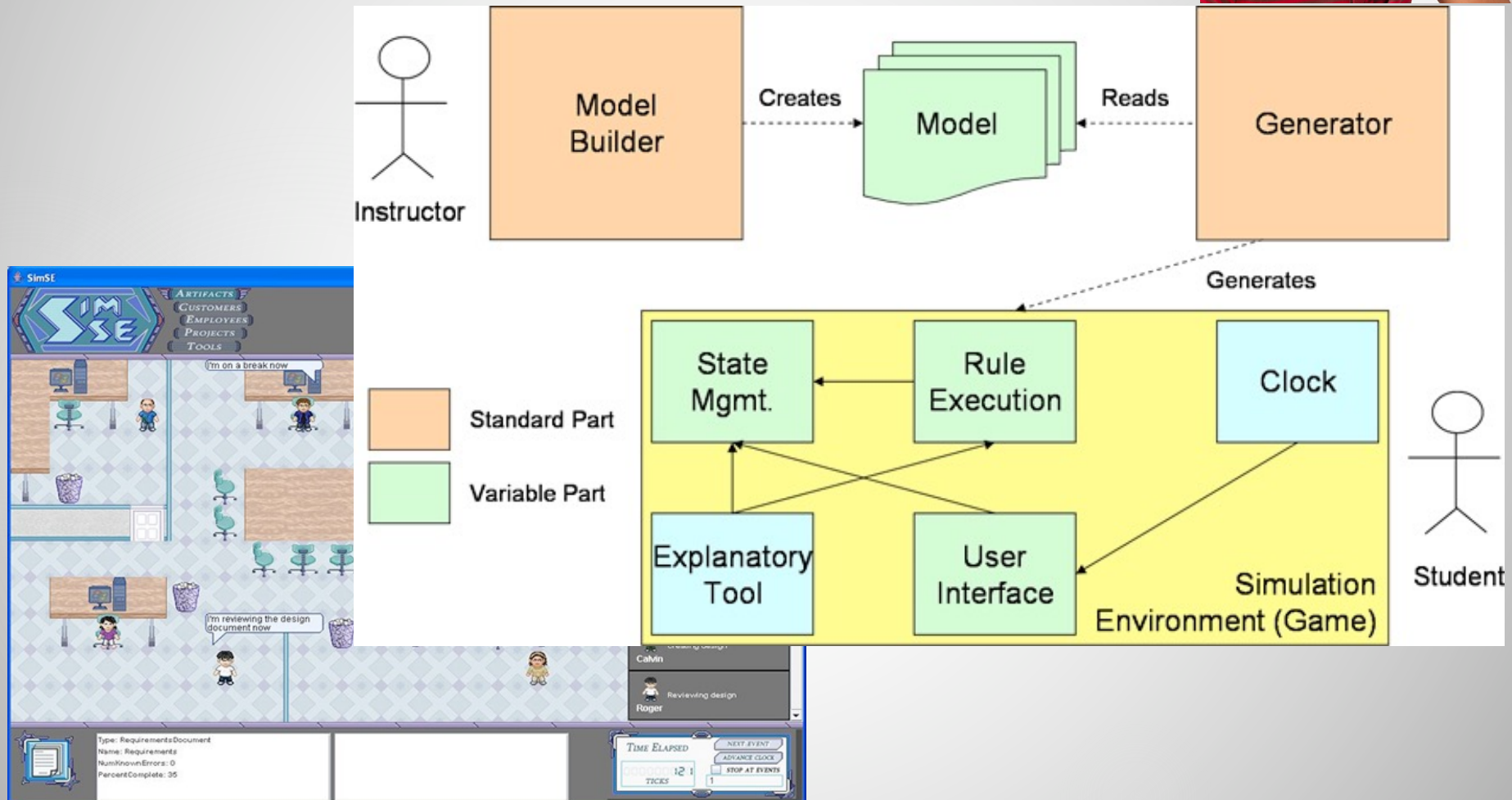
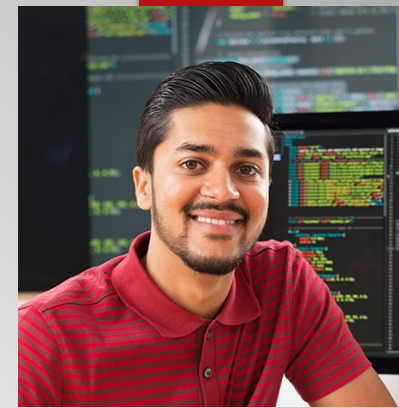
<http://spectrum.ieee.org/cars-that-think/transportation/self-driving/the-scary-efficiency-of-autonomous-intersections>

# Andy's Perspective: Medical Systems

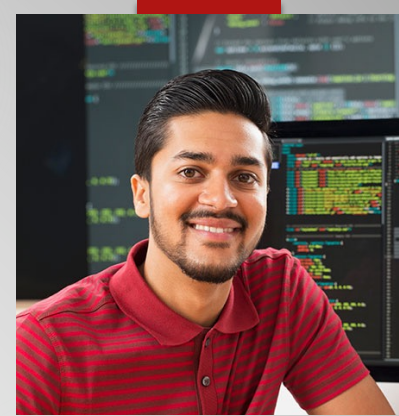




# Andy's Perspective: Video Games

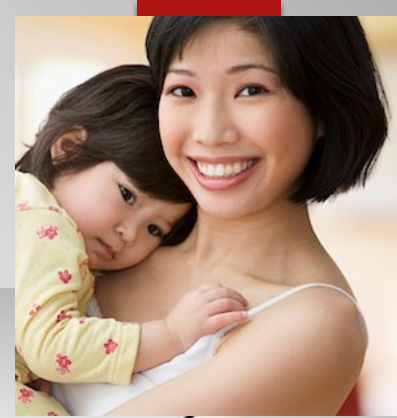


# Andy = **Engineering** perspective



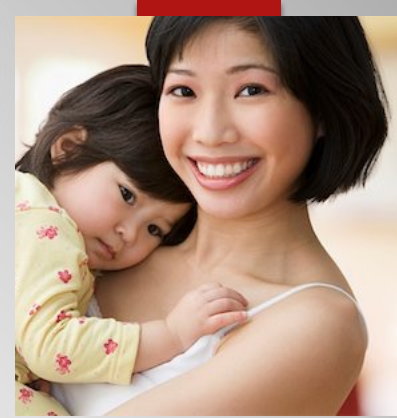
- What matters most?
- How does software engineering help?

# Clara's Perspective: Cars





# Clara's Perspective: Medical Systems



“Unfortunately, Ava is extremely **not awesome looking**....

They should take a hint from Apple and make her **look awesome.**”

# Clara's Perspective: Video Games



“...the things that make a boss battle **boring** are, "stupid amounts of **repetition**, ridiculously high/replenishing energy [i.e. boss health] combined with **unimaginative gameplay** (yawn), and powered up versions of previous bosses.”



Sources: [http://www.designersnotebook.com/Design\\_Resources/No\\_Twinkie\\_Database/no\\_twinkie\\_database.htm](http://www.designersnotebook.com/Design_Resources/No_Twinkie_Database/no_twinkie_database.htm)

# Clara = **User** perspective



- What matters most?
- How does software engineering help?

# What parts of software do we “see?”

- *User*: the user interface – buttons, icons, menus, etc.
- *Programmer*: source code, internal design
- *Electrical engineer*: semiconductors, transistors, power supply
- *Society*: impact, output, trends, dangers
- *Business people*: profits, costs, required training
- *Designers*: product, structure
- *Software project manager*: teams, budgets, timetables
- *Professor of software engineering*: principles, theories, models
- *Students of software engineering*: ~~work, work, work~~ opportunity, challenge, fun

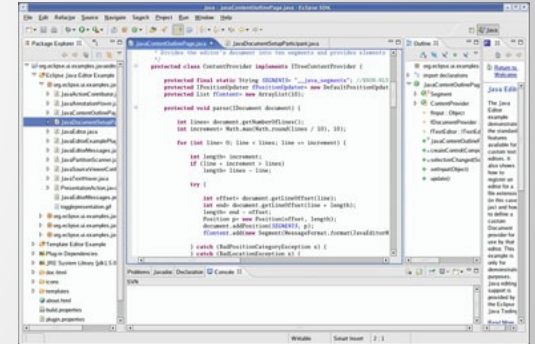
# Today's Lecture – What is software engineering?

- Perspectives on software engineering
- “Essential ingredients”
- Software Principles
- No Silver Bullet
- Quiz 1 study guide



# Three “Essential Ingredients” of Software Engineering

**People** + **Processes** + **Tools**



# Today's Lecture – What is software engineering?

- Perspectives on software engineering
- “Essential ingredients”
- **Software Principles**
- No Silver Bullet
- Quiz 1 study guide

# Principles of software engineering – Davis, 1994

- Make quality number 1
- Give products to customers early
- Understand the problem first
- Choose an appropriate process
- Good management is more important than good technology
- People are the key to success

# Principles of software engineering – Royce, 1998

- Follow an architecture-first process
- Use component-based development (buy vs. build) to reduce the coding effort
- Show the customer preliminary versions of the software frequently
- Have incremental releases

# Principles of software engineering – Wasserman, 1996

- Modularity
- Abstraction
- User interface (UI) prototyping
- Reuse
- Metrics



# Inf 43 Recurring, fundamental principles

- Rigor and formality
- Separation of concerns
  - modularity
  - divide and conquer
  - abstraction
    - <https://www.youtube.com/watch?v=p7nGcY73epw>
- Anticipation of change
- Generality
- Incrementality



# Today's Lecture – What is software engineering?

- Perspectives on software engineering
- “Essential ingredients”
- Software Principles
- No Silver Bullet
- Quiz 1 study guide

# No Silver Bullet – Essence and Accidents of Software Engineering, by Frederick Brooks



# Essential properties of software

- Complexity
- Conformity
- Changeability
- Invisibility



# Complexity

- “...a construct of interlocking concepts: data structures and classes, algorithms, function calls.”
- “No two parts are alike.”
- Consequences of this?

# Conformity

- Software must conform to human institutions and systems

# Changeability

- “All successful software gets changed.”
- Why?

# Invisibility

- “The reality of software is not inherently embedded in space.”
- What are the consequences of this?

# “False” Silver Bullets



- High-level languages
- Time-sharing
- Unified programming environments
- Object-oriented programming
- Artificial intelligence
- Expert systems
- Automatic/graphical programming
- Program verification
- Environments/tools
- Workstations

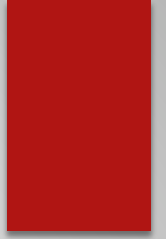


# Potential Silver Bullets

(Not *really* "silver bullets," but these were thought that they might greatly help)

- Buy vs. build
- Requirements refinement and rapid prototyping
- Incremental development
- Great designers

# Attendance Quiz



# Summary (I)



- Software engineering can be looked at from different perspectives
  - Business
  - Engineering
  - User
  - Others
- Essential ingredients of software engineering
  - People + Processes + Tools
- Inf 43 principles of software engineering
  - Rigor/formality, separation of concerns (modularity, divide and conquer, abstraction), anticipation of change, generality, incrementality

# Summary (II)

- No Silver Bullet
  - Software engineering is hard because of the essential difficulties of software
    - Complexity, conformity, changeability, invisibility
  - Potential ideas (at the time) that could *help*, but not completely solve
    - Buy vs. build
    - Requirements refinement/rapid prototyping
    - Incremental development
    - Great designers



# Today's Lecture – What is software engineering?

- Perspectives on software engineering
- “Essential ingredients”
- Software Principles
- No Silver Bullet
- Quiz 1 study guide

# Quiz 1 – Basics

- Tuesday, Oct. 10, 4:20–4:50pm
  - You must take it during this time
- Online Canvas quiz (named “Quiz 1”)
  - Open-notes, open-book
  - Part multiple-choice, part free-form
    - I will tell you how many of each in the quiz instructions so you can budget your time accordingly
  - A TA will be in a chat room to answer any questions via text. Everyone will be muted and no video. I will share the link on Canvas in the schedule.
- **No collaborating—take it individually**

# Quiz 1 – How to Study



- Different from other CS courses
  - Software engineering as much about people as it is about software
  - Shifts away from technical thinking of a CS student
  - Many ways to analyze topics, especially definitions, links between different concepts
- Attend lecture, take notes, spend time going over them carefully, analyzing, discussing
- Do readings carefully, take notes, analyze, and discuss

# Quiz 1 – Topics (I)

- Be able to explain each quoted definition of software engineering, especially the **key terms**
- 3 essential ingredients of software engineering
- Know and understand the 3 perspectives on software engineering we talked about (and which is considered to be the most important)
- Know and understand the “Inf43 Recurring, Fundamental Principles” of software engineering



# Quiz 1 – Topics (II)

- No Silver Bullet
  - Know and understand (be able to explain) the essential difficulties of software engineering
- Make sure you have watched the videos shown in class
- **The quiz will focus on these topics, but I reserve the right to ask about any other lecture/reading information as well**

# To do + Next Time

- **To Do:**
  - Take the attendance quiz by tomorrow at 11:59pm
  - Complete assigned readings before attending Tuesday's lecture
    - Catastrophic failures caused by faulty software
  - Study for Quiz 1
- Quiz 1 Tuesday 4:20–4:50pm
- Next lecture:
  - Software failures
  - Requirements engineering