CSC 436, Fall 2017

Software Engineering

Ravi Sethi

THE UNIVERSITY OF ARIZONA

Version: 1

August 21, 2017

CSC 436: Software Engineering

THE UNIVERSITY OF ARIZONA
Computer Science

Piazza Link https://piazza.com/arizona/fall2017/csc436/home

- **Professor: Ravi Sethi**
  - rsethi@email.arizona.edu
  - 520-621-0689
  - Gould-Simpson 720
  - Office Hours:
    Tue & Thu 10:00-11:00
    or by appointment
    (Subject to Change)

- **TA: Jacob Combs**
  - jacobcombs@email.arizona.edu
  - 512-822-1484
  - Gould-Simpson 938
  - Office Hours:
    Mon & Wed 1:00-2:00
    or by appointment
    (Subject to change)

Your "Program and Plan" includes

In addition to Computer Science

- Aerospace
- Anthropology
- Business Adminstration
- Chinese Studies
- Communication
- Creative Writing
- Electrical and Computer Engineering
- Film and Television
- Information Science Technology & Arts
- Mathematics
- Molecular & Cellular Biology
- Music
- Physics
- Studio Art

On the index cards please write

- On the small card
  - What would you prefer to be called?
  - At the top of the card, please write your preferred name and last name
- On the large card
  - What would you like to get from this course
  - This can be anonymous – you need not identify yourself

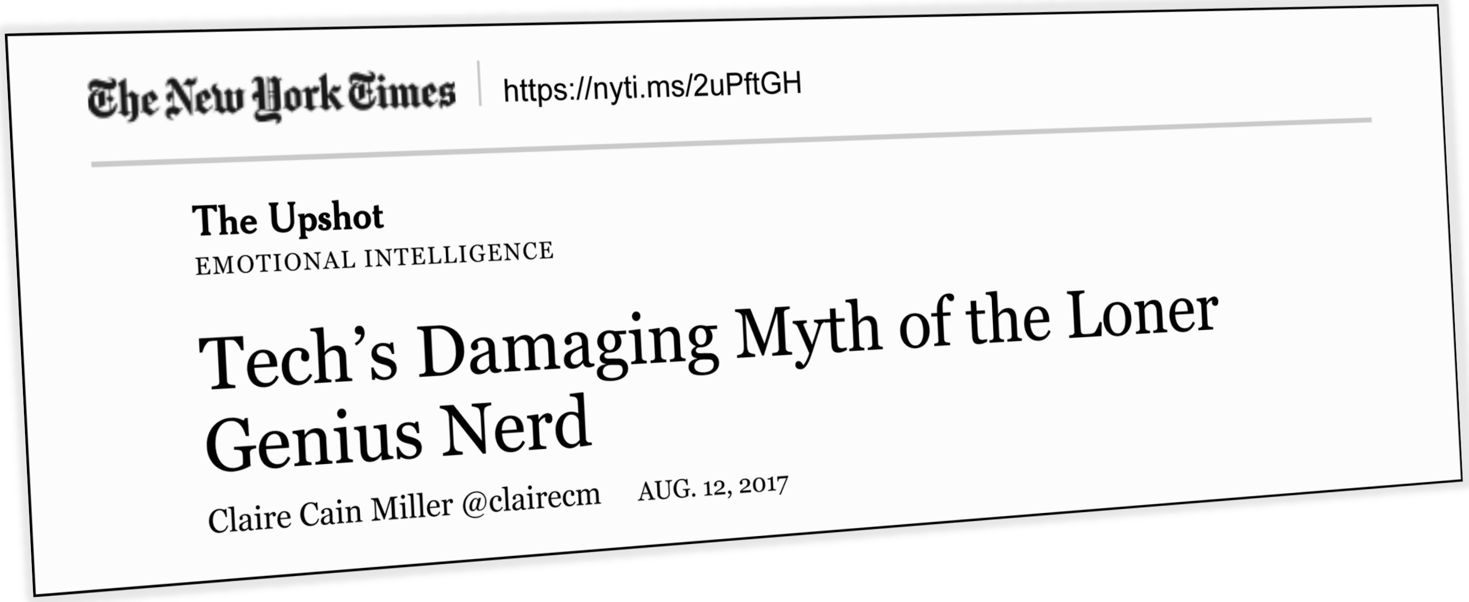
About CSC 436: Software Engineering

Welcome!

- **Emphasis: principles and practices**
  - With real-world examples of what's worked
- **Team projects are central to the experience**
  - Form your own teams of 4 for the semester
  - Interim reports and reviews will help you stay on track

What do Software Engineers do?

THE UNIVERSITY OF ARIZONA
Computer Science



The New York Times | https://nyti.ms/2uPftGH

The Upshot
EMOTIONAL INTELLIGENCE

Tech's Damaging Myth of the Loner Genius Nerd

Claire Cain Miller @clairecm AUG. 12, 2017

Software Engineers/Developers: Duties

THE UNIVERSITY OF ARIZONA.
Computer Science

What do Software Engineers do? A view …

- **Qualities and Skills**
  - Analytical, Creativity, Problem Solving
  - Communication, Customer-Service, Inter-Personal
  - Big Picture, Attention to Detail

- **Software developer jobs, 2014: 1,114,000**
  - Applications developers 64%, systems developers 36%
  - Employment projected to grow 17% (2014-2024), much faster than the 7% average for all occupations
  - Median annual pay, 2016: $102,280

Source: U. S. Bureau of Labor Statistics

Resources on Piazza

Please enroll, if you haven't already

- **Course Materials**
  - Announcements
  - Project Information
  - Lecture Slides
  - Homework
- **Draft Textbook**
  - Posted: Chapters 1-7, 9-11
  - The draft will be revised during the course
  - I'd appreciate comments and corrections

Software Engineering

Ravi Se

The University o

January 4,

© 2017 Ravi Sethi. This working draft is intended
436, *Software Engineering*, at the University of Ar

Contents

v

What is Software Engineering?

“a concise and complete definition of software engineering is difficult to formulate”

Source: IEEE and ACM “Software Engineering 2014” Curriculum Guidelines

Source: NATO Conference on Software Engineering (1968)

State of the Art in 1968

From the NATO Conference

"We build systems like the Wright brothers built airplanes—build the whole thing, push it off the cliff, let it crash, and start over again."

Source: NATO Conference on Software Engineering (1968)

State of the Art in 1968

From the NATO Conference

"We build systems like the Wright brothers built airplanes—build the whole thing, push it off the cliff, let it crash, and start over again."

In 1968, "the phrase 'software engineering' was deliberately chosen as being provocative, in implying the need for … the types of theoretical foundations and practical disciplines that are traditional in established branches of engineering"

Source: NATO Conference on Software Engineering (1968)

THE UNIVERSITY OF ARIZONA
Computer Science

Distinction between Process and Product

Process	Product
who does what by when	what gets developed
teams and organization	systems and artifacts

Software Engineering

Dimensions

Customers

Stakeholder Needs

Requirements

Teams

Processes

Goals & Metrics

Art & Science

Principles

Practices

Technology

Systems

Tools

Context

Constraints, Cost, Schedule

Business, Legal, Regulatory

Software Engineering

Working definition, for this course

- Software engineering is
  - the art and science of
  - developing reliable software systems that
  - address customer needs,
  - subject to resource, business, and societal constraints.

Software Eng.: Alternative Definition

“multi-person development of multi-version programs”

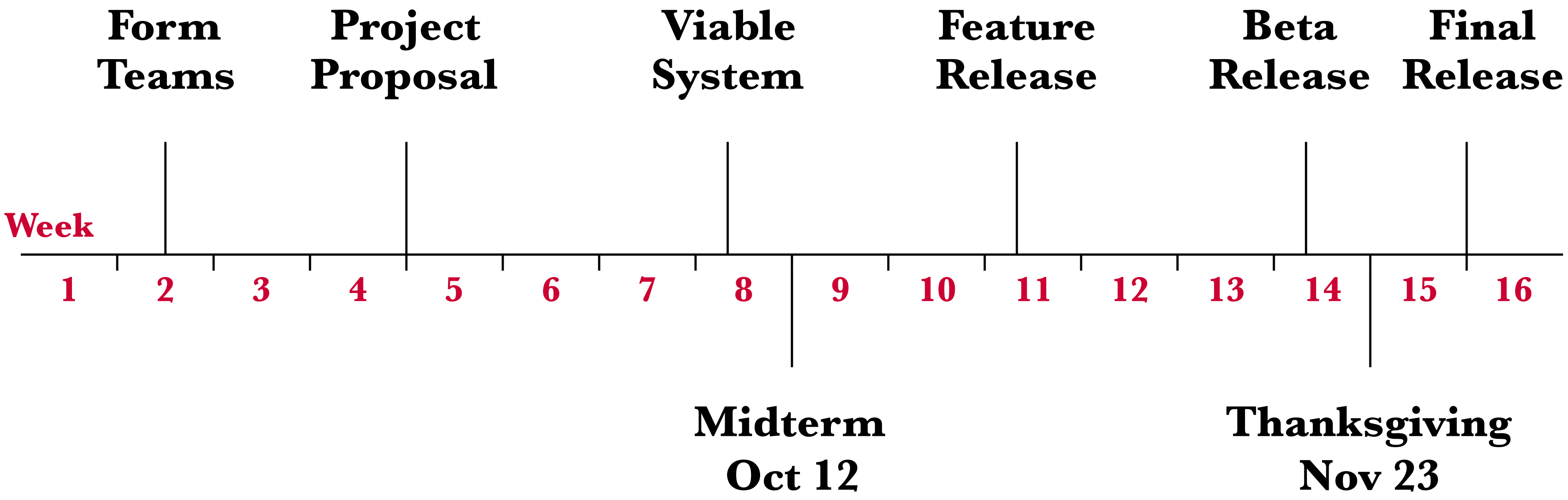
Multi Person Coordinate Teams Design for Modularity	**Multi Person** **Multi Version**
Single Person	**Multi Version** Develop Program Families Evolve & Maintain Releases

Source: attributed to Brian Randell by Parnas [2011]

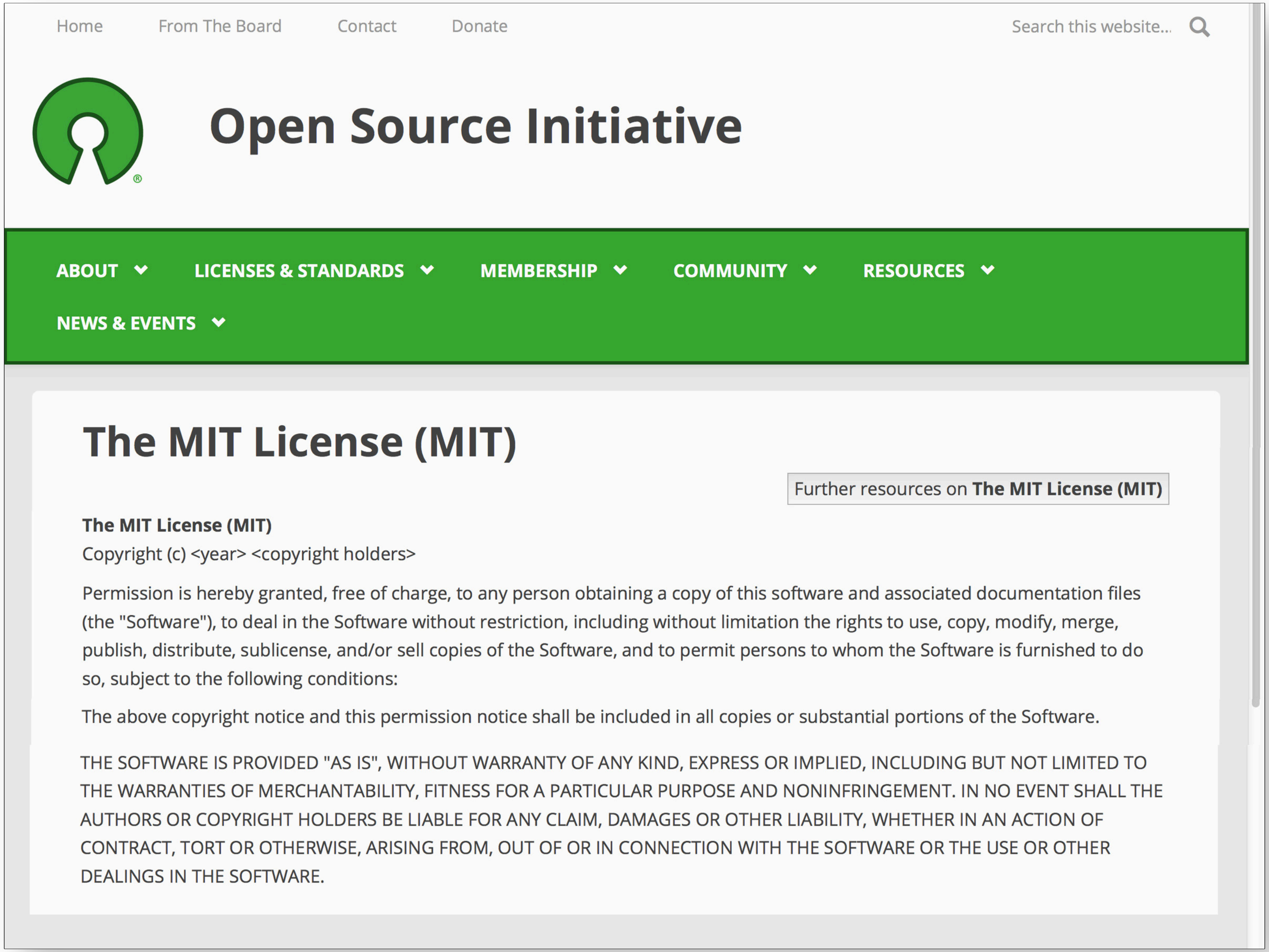
Team Project

CSC 436: Rough Timeline

Project



Software and Docs Provided "As Is"



The MIT License (MIT)

The MIT License (MIT)

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

http://opensource.org/licenses/MIT

CSC 436, Fall 2017: Policies

See the Syllabus

CSC 436: Grading Policy

Computer Science

UA Grading Policy: http://registrar.arizona.edu/gradepolicy

- **Homework** 20%
  - Homework submissions are individual
- **Team Project and Reports** 40%
  - 5% of the 40% is individual; 35% same for all team members
- **Exams** 40%
  - Equal weight for Mid-Term and Final
- **Instructor Discretion (bonus points)** up to 5%
  - Closer to 0% than 5%

CSC 436: Grading Distributions

THE UNIVERSITY OF ARIZONA
Computer Science

Any revisions to these buckets will be on the generous side

A	90+%
B	80 - 89%
C	70 - 79%
D	60 - 69%
E	0 - 59%

CSC 436: University Attendance Policy

Computer Science

Full policy at http://catalog.arizona.edu/2015-16/policies/classatten.htm

- **Excerpts from the University Attendance Policy**
  - "Students are expected to be regular and punctual in class attendance and to fully participate in the course."
  - "Excessive or extended absence from class is sufficient reason for the instructor to administratively drop the student from the course."
- **Policy for Religious Holidays**
  - http://deanofstudents.arizona.edu/policies-and-codes/accommodation-religious-observance-and-practice

UA Computer Science Code of Conduct

THE UNIVERSITY OF ARIZONA
Computer Science

Full policy at http://www.cs.arizona.edu/codeofconduct.pdf

We, the students and professionals of the UA Department of Computer Science, are committed to providing and maintaining a supportive community and a thriving educational environment.

We strive to:

- Be Welcoming and Inclusive
- Honor Privacy and Confidentiality
- Continue to Improve Our Learning Environments
- Behave Respectfully and Courteously
- Demonstrate Intellectual Honesty

Thank you for your contributions in support of these goals. Disruptive behaviors such as physical or emotional harassment, dismissive attitudes, and misuse of department resources are contrary to this code. If you have questions, want to make us aware of a problem, or wish to see someone recognized for their positive actions, please email: depthead@cs.arizona.edu

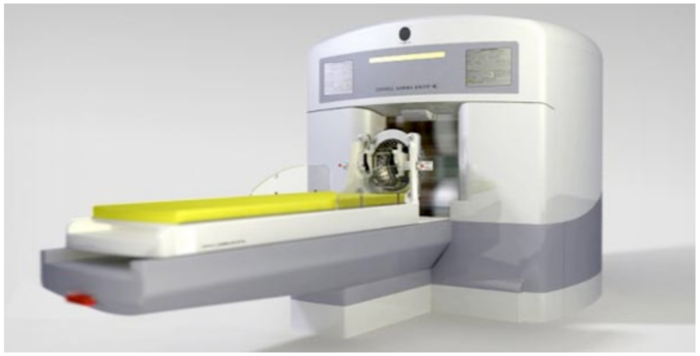
Ethics

"Because of their role in developing software systems, software engineers have significant opportunities to do good or cause harm."

Source: ACM Software Engineering Code of Ethics, https://www.acm.org/about/se-code

Therac-25: Fatal Radiation Overdose

A patient died



- **6 known accidents 1985-1987**
  - Series of "fixes"
  - "Inadequate software engineering practices"
- **E. Texas Cancer Center, March 21, 1986**
  - Therac-25 had been in use for 2 years; over 500 patients treated.
  - Male patient, in for his ninth treatment
  - Planned dose: 180 rads
  - Possible dose: 16,500-25,000 rads in less than 1 sec.
  - 5 months later, the patient died from complications of the overdose

Sources: Leveson and Turner [1993], Leveson [1995]

Ethics

From the ACM Code of Ethics and Professional Conduct

- Contribute to society and human well-being
- Avoid harm to others
- Be honest and trustworthy
- Be fair and take action not to discriminate
- Honor property rights including copyrights and patent
- Give proper credit for intellectual property
- Respect the privacy of others
- Honor confidentiality

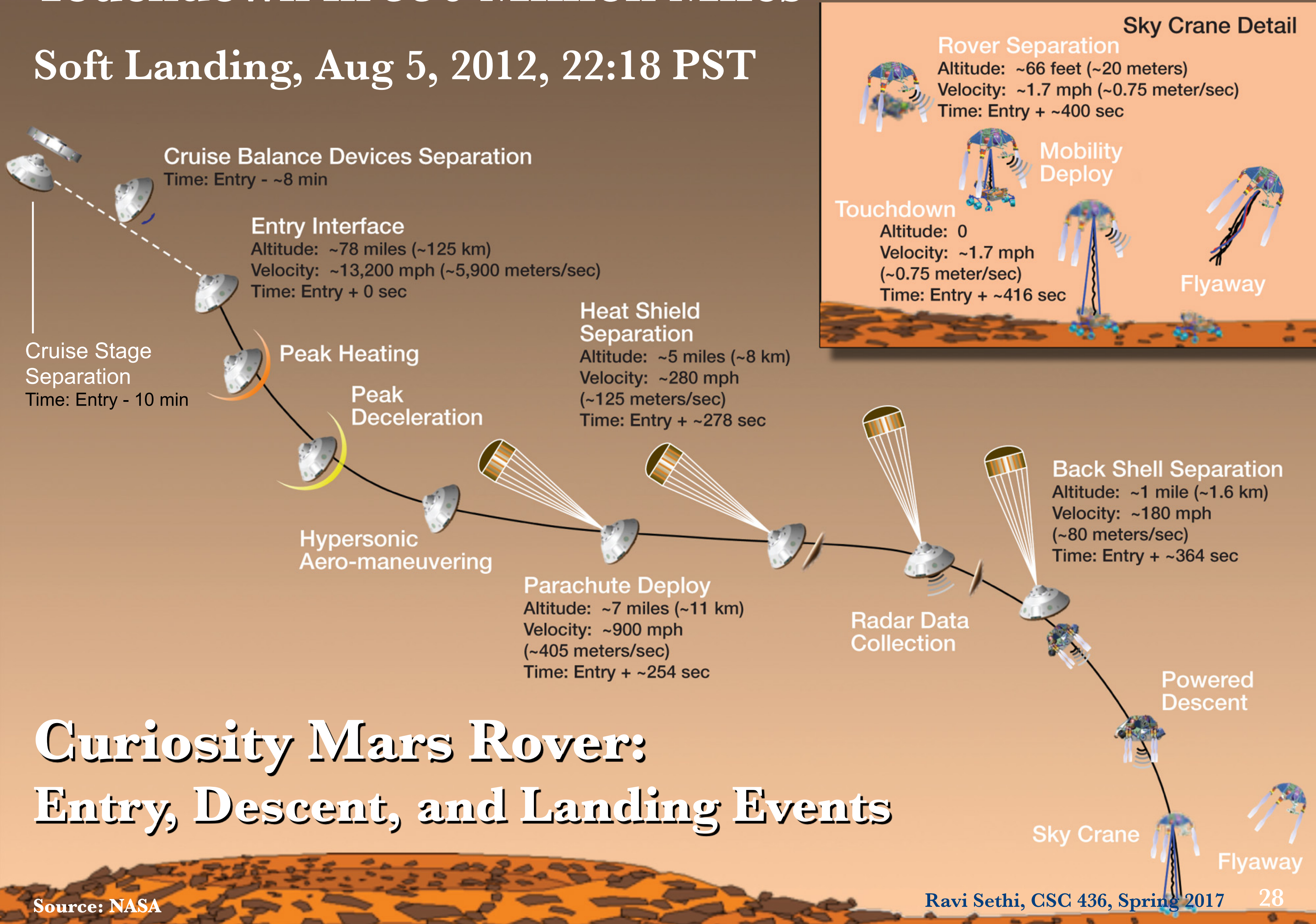
Source: ACM Software Engineering Code of Ethics, https://www.acm.org/about/se-code

Software Engineering Success Story

Landing the Rover, Curiosity, on Mars

Touchdown in 350 Million Miles

Soft Landing, Aug 5, 2012, 22:18 PST



Curiosity Mars Rover: Entry, Descent, and Landing Events

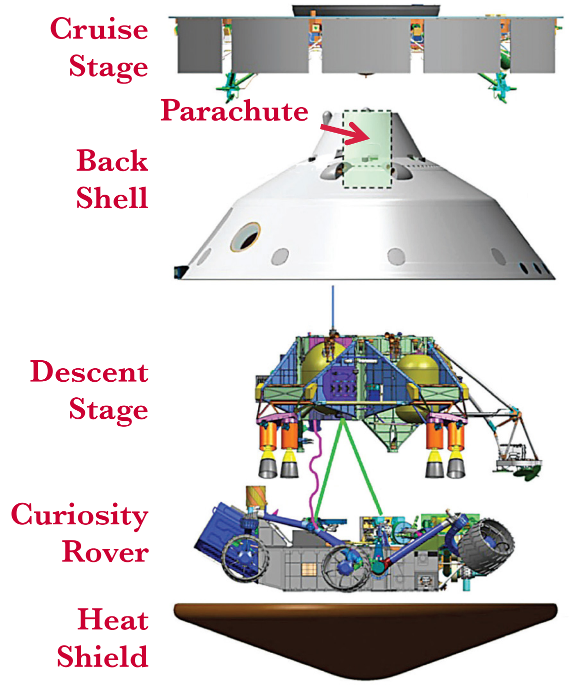
Source: NASA

Landing the Curiosity Rover on Mars

THE UNIVERSITY OF ARIZONA
Computer Science

All functions on rover & spacecraft controlled by software

- Curiosity (900 kg) was lifted by an Atlas V 541 rocket (531,000 kg)
- Within two hours of launch all that remained were the systems on the right
- The solar panels on the Cruise Stage powered the craft on the 9-month journey to Mars
- 10 min. to landing: cast off Cruise Stage
- 3 min. to landing: deploy Parachute to slow from 1,500 km/hour to 300 km/hour
- <1 min. to landing: Descent Stage separates, along with the Rover
- The Descent Stage must now guide the Rover, disconnect, and fly away a safe distance to crash



Source: Holzmann [2014]

Backup

Student Comments from Past Projects

From the Reflection section of a Team's Final Report

"We all understood that a successful project could really help in the impending hiring process so we really wanted to make something we felt proud of and confident in presenting to prospective employers."

Student Comments from Past Projects

THE UNIVERSITY OF ARIZONA. Computer Science

From the Reflection section of a Team's Final Report

"Make sure you have a thorough understanding of the requirements your customer has before you begin, otherwise you may end up doing two or three times the necessary work due to changing requirements."

Software for the Flawless Mars Landing

Good tools, workmanship, careful process

- **Flight-control software on the spacecraft**
  - 35 developers; 3M lines of C, some C++; 75% generated
- **Prevention, learning from problems on previous missions**
  - Coding rules, with mechanical compliance checks; e.g., static analysis
  - Developer certification, emphasizing software risk & defensive coding
- **Detection: nightly builds, with checkers running in parallel**
  - Peer reviews excel at finding design flaws, not at compliance checks
  - Verification (model checking) for possible race conditions or deadlocks; in one case, findings led to the complete redesign of a subsystem
  - Rigorous unit and system testing; full code coverage (extreme requirement)
- **Containment of residual defects within a module**
  - For the critical landing sequence, the backup CPU ran a stripped-down version

Source: Holzmann [2013, 2014]

Landing the Curiosity Rover on Mars

Overall a successful mission, but over budget and late

- First call for scientific instruments 4/04
- Instruments approved, design of mono propellant engine begins 12/04
- "Confirmation Review" … "cutting hardware" begins 8/06
- "Critical Design Review" (CDR) signaled problems 6/07
  Removed grinder, zoom lens, asked for $75M additional
- Major review by JPL and independent technical review team 10/08
  Decided doable by 2009 with launch 3 weeks later and $200M to speed testing
  Set weekly milestones for Nov-Dec; review in Jan whether to spend the $200M
- Delay Decision: Missing milestones 12/08
  Decision to delay launch until 8/11
- Flawless landing on Mars on August 6, 2012, but
  $900M over budget and 24 Months late