SE491/591: SOFTWARE ENGINEERING STUDIO

Jane Cleland-Huang
Office: CDM 836
Hours: Tuesday and Thursday 11.00am-12.30pm
or upon request
Phone: 312-362-8863
Email: jhuang@cs.depaul.edu

DePaul University

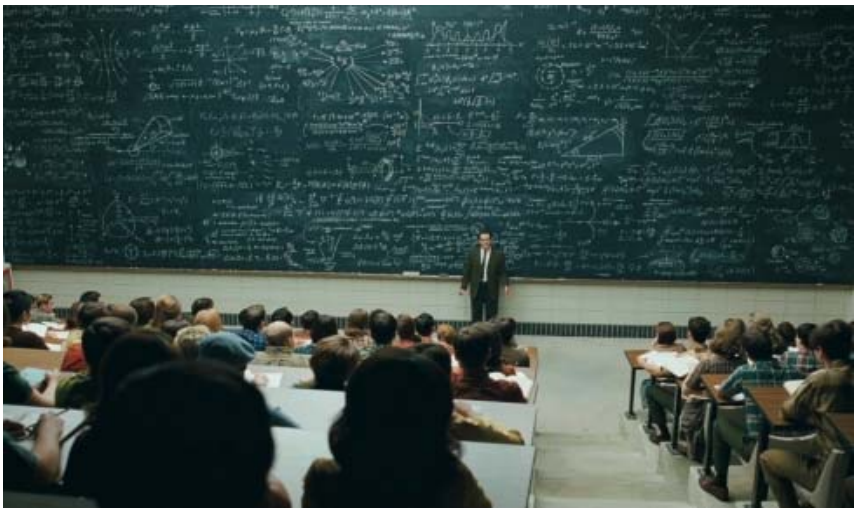
CoEST
Center of Excellence for Software Traceability

What are the goals of Studio?

- To provide students with experience in developing a large, software project;
- To work as part of a cooperating team
- To produce substantive intermediate deliverables within realistic time and resource constraints
- To expose students to appropriate development processes and environments.
- To have students produce a fully-functional, fully-featured final product.

2

Studio is NOT a lecture-based course



3

Skills “may” involve the following:

- Use of Java and Java-related technologies including advanced programming techniques;
- Android and iOS development
- Cloud computing
- Object-oriented analysis and design skills
- Software architectural design;
- Unit and integration testing;
- User interface design;
- Use of development methodologies such as the Unified Process and/or selected Agile development methods;
- Application of project management skills and techniques;
- Team collaboration

4

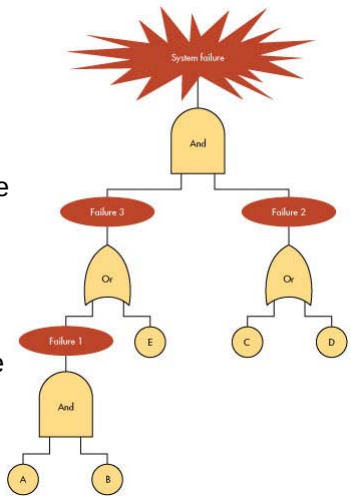
In this Offering of Studio

- Physical computing projects – interacting with sensors and/or actuators
- Quasi safety-critical systems.
  - Impacts the tools and development processes we will use
  - Impacts the `artifacts' we will produce
  - Focus is on delivering executable, functioning, **safe-for-use** code.
- **Each product** will involve mobile apps, web-services, physical computing.
- Teams will all have 5-6 people to manage the workload.

5

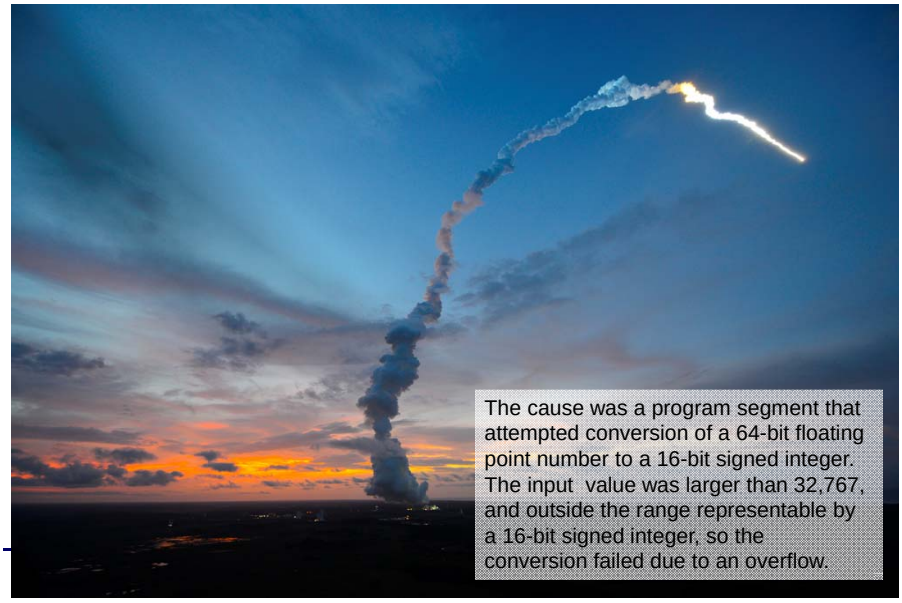
Safety Critical Systems

- Safety-critical systems are systems that can cause harm or catastrophic damage if they fail.
- They include medical devices, automotive braking systems, nuclear power plant controls, and avionic flight management systems.
- Most safety-critical systems must be certified by a regulatory agency to ensure that they are fit-for-purpose.
- Hazards often include interactions between hardware and software.



6

Ariane 5



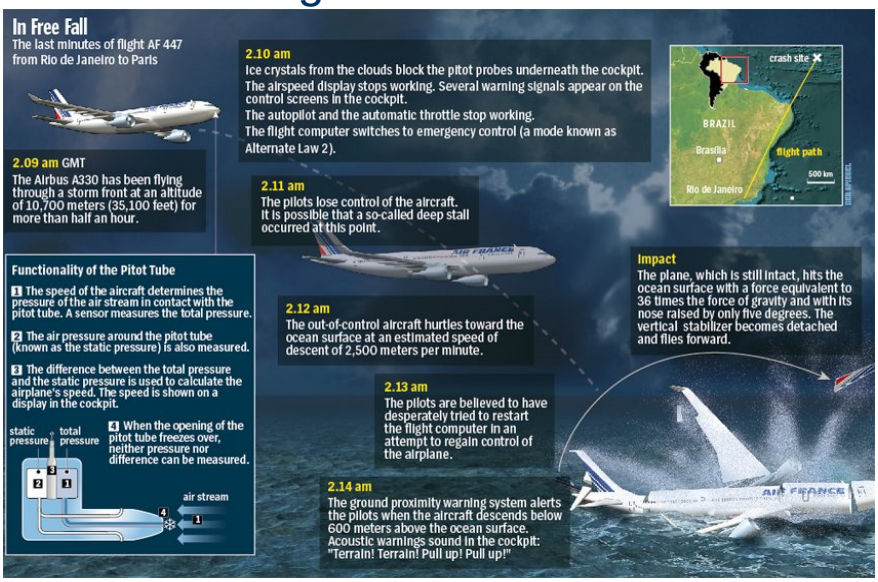
The cause was a program segment that attempted conversion of a 64-bit floating point number to a 16-bit signed integer. The input value was larger than 32,767, and outside the range representable by a 16-bit signed integer, so the conversion failed due to an overflow.

Hatch Nuclear Power Plant

The Edwin I. Hatch nuclear power plant was rforced into an emergency shutdown for 48 hours after a software update was installed on a computer. The software update was designed to synchronize data on both the business system computer, and the control system computer. According to a report filed with the Nuclear Regulatory Commission (NRC), when the updated computer rebooted, it reset the data on the control system, causing safety systems to errantly interpret the lack of data as a drop in water reservoirs that cool the plant's radioactive nuclear fuel rods.

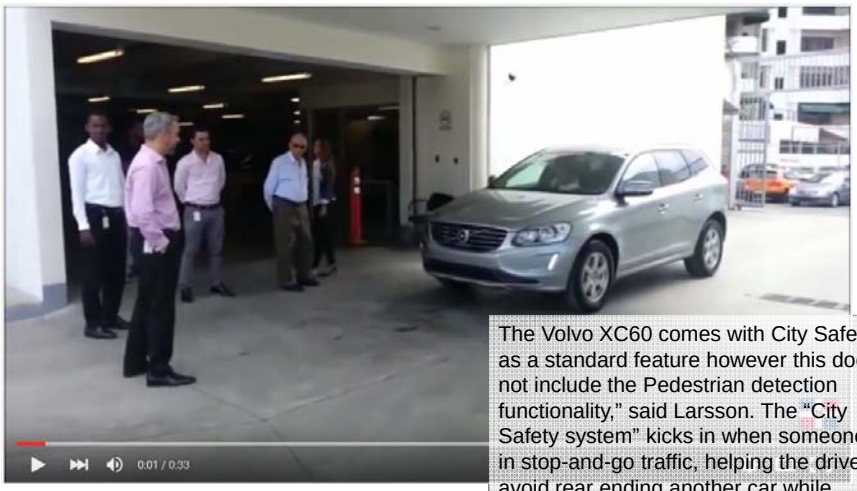
8

Air France Flight 447



Software issues: The onboard automating reporting system transmitted 24 sometimes conflicting error messages regarding discrepancies in the indicated air speed (IAS) readings before the aircraft disappeared.

Self-Driving Car Accident

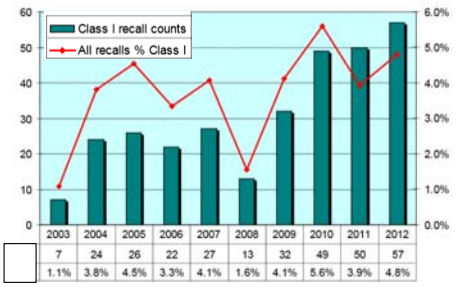


https://www.youtube.com/watch?v=_8nnhUCtcO8

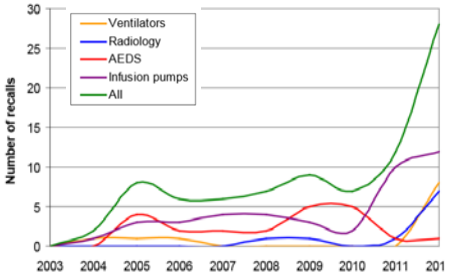
The Volvo XC60 comes with City Safety as a standard feature however this does not include the Pedestrian detection functionality," said Larsson. The "City Safety system" kicks in when someone is in stop-and-go traffic, helping the driver avoid rear ending another car while driving slowly, or under 30 mph.

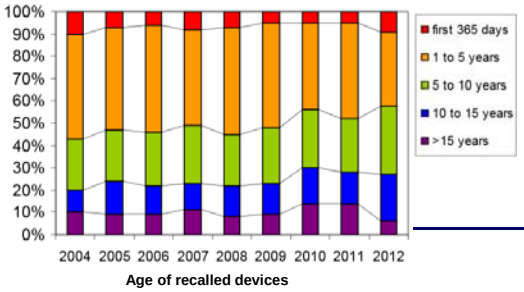
10

Medical Device Recalls



	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
	7	24	26	22	27	13	32	49	50	57
	1.1%	3.8%	4.5%	3.3%	4.1%	1.6%	4.1%	5.6%	3.9%	4.8%





The most frequent causes for recalls are related to device design, software, and non-conforming material or component issues.

11

Project Domains and Ideas

Drones	E-Health	Environment	Mission-Control
			
Fleets of drones are coordinated to deliver medical supplies in a natural catastrophe and to use aerial reconnaissance for tracking.	A person recovering from a medical issue such as a heart attack, is safely monitored during his/her rehabilitation.	Crowd-sourced pollution detection in case of a chemical spill. (Also for environmental pollution).	Soldiers (or rescue team) on a mission. Monitor position & health of team. Provide instruct-ions and informa-tion via visor.

12

Project: Medical Drones

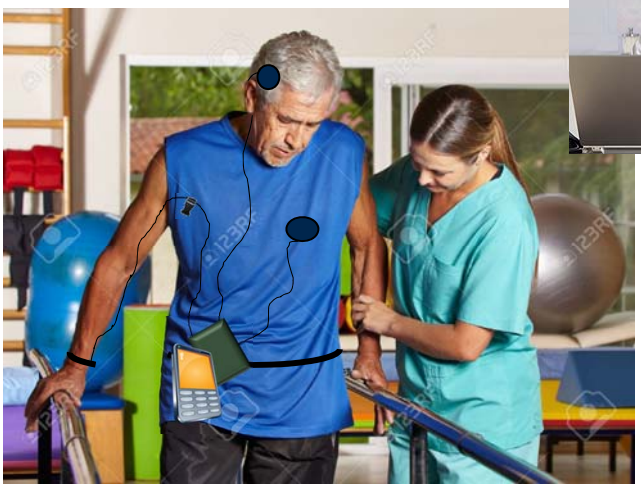
Drone supply base

Request for help

Central controller, dispatch algorithms, drone tracker.

13

Project: Rehabilitation



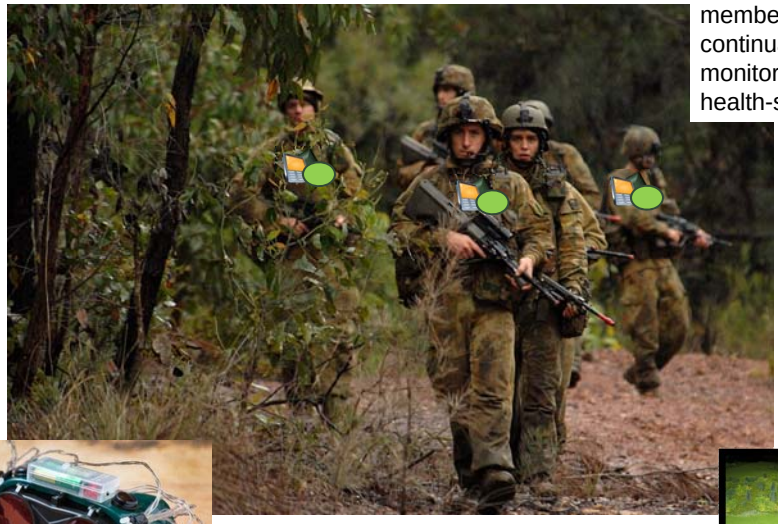
Doctor or therapist creates a safe exercise plan for the patient.

Exercise sessions are monitored and data is tracked via the mobile to the server.
You can also include outside exercise and GPS tracking.

14

Project: Mission Possible

Health of team members continually monitored using health-sensors.

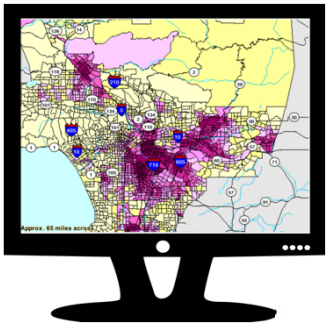


Centralized control tracks mission and plans route.

Group dynamics, route plan, and instructions displayed on glasses.

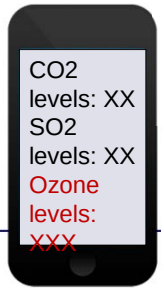
15

Project: Environmental Disaster



Web-server captures information about pollutants (aka dangerous chemicals) and displays them on a map.

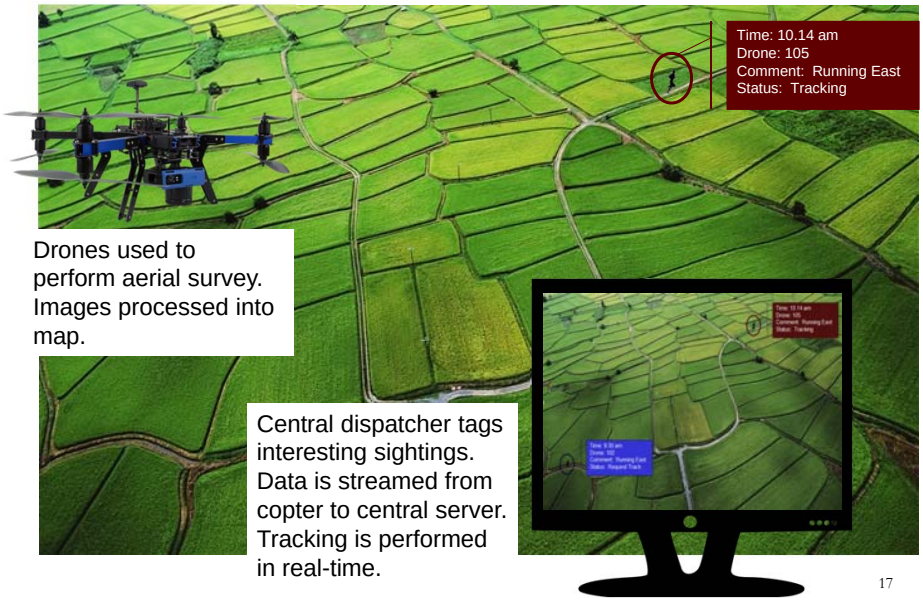
Crowd-source the **collection** process.



Ordinary users can download an App that warns them if they approach unsafe areas.

16

Project: Aerial Reconnaissance

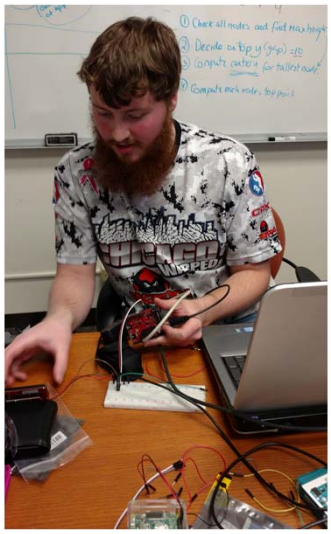


Drones used to perform aerial survey. Images processed into map.

Central dispatcher tags interesting sightings. Data is streamed from copter to central server. Tracking is performed in real-time.

17

Our very own "Q"



Robert Cole has tested all the interfaces to physical devices and created and collected instructions for how to send commands and/or extract data from the devices.

We will distribute physical devices and hardware to each team in Week 3.

At that time, Robert will meet with teams to explain how to control and/or to retrieve data from your device.

He has also set-up instructions here.

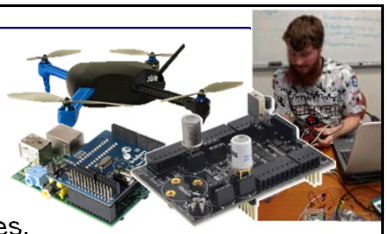
18

Scientific Research

- We are funded by the National Science Foundation http://www.nsf.gov/awardsearch/showAward?AWD_ID=1513730&HistoricalAwards=false to investigate Traceability across Safety Critical Product lines.
- High recall/fault error occurs when safety-critical devices such as medical devices are modified (i.e. new features added).
- Recreating a safety case is time-consuming and error-prone.
- We are engaging in fundamental research to develop techniques for evolving trace links across safety-critical products.
- These techniques can potentially lead to automated traceability, and meaningful guided domains for constructing safety-cases.

19

Please Participate



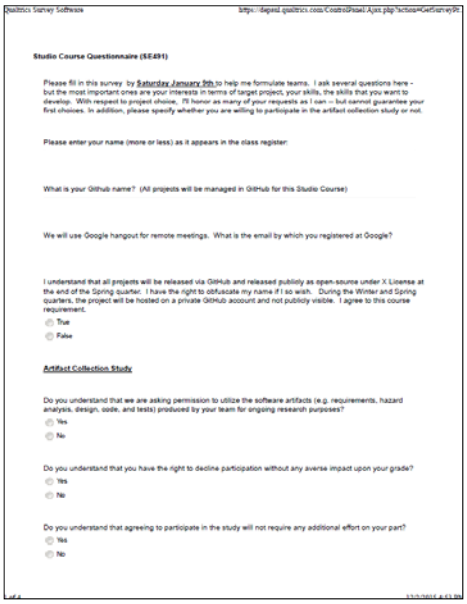
- Please consider giving your permission to use the requirements, code, test-cases etc. that you produce during Studio to support future traceability studies.
- **No extra effort required**
  Everyone in Studio – regardless of their participation will perform the same work, create the same artifacts, use the same tools etc.
- **Reward**
  In recognition of your contribution – at the end of each of the two quarters we will (during class) randomly select two study-participants to win $50 Amazon tokens.
- **Impact**
  There will be no negative impact upon your grades if you choose not to participate. However, for practical purposes, we need to separate teams into participants and non-participants.

All equipment for the Studio course is funded entirely by Research funds. However, no NSF funds have been used.

20

How do I Sign up?

- **Everyone** needs to complete a Studio survey by Sunday 10th, January.
- This is a **4-page survey** which primarily focuses on your preferences for projects, team composition, team meetings etc.
- About 4 questions concern your participation in this study. If you agree to participate you will also need to sign the IRB consent form.



Studio Course Questionnaire (SE491)

Please fill in this survey by Saturday January 9th to help me formulate teams. I ask several questions here - but the most important ones are your interests in terms of target project, your skills, the skills that you want to develop. With respect to project choice, I'll honor as many of your requests as I can -- but cannot guarantee your first choices. In addition, please specify whether you are willing to participate in the artifact collection study or not.

Please enter your name (more or less) as it appears in the class register:

What is your Github name? (All projects will be managed in GitHub for this Studio Course)

We will use Google hangout for remote meetings. What is the email by which you registered at Google?

I understand that all projects will be released via GitHub and released publicly as open-source under X License at the end of the Spring quarter. I have the right to obfuscate my name if I so wish. During the Winter and Spring quarters, the project will be hosted on a private GitHub account and not publicly visible. I agree to this course requirement.

- True
- False

Artifact Collection Study

Do you understand that we are asking permission to utilize the software artifacts (e.g. requirements, hazard analysis, design, code, and tests) produced by your team for ongoing research purposes?

- Yes
- No

Do you understand that you have the right to decline participation without any averse impact upon your grade?

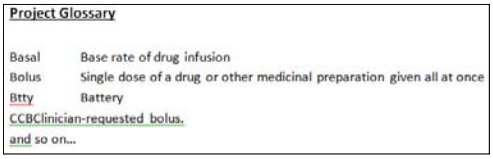
- Yes
- No

Do you understand that agreeing to participate in the study will not require any additional effort on your part?

- Yes
- No

21

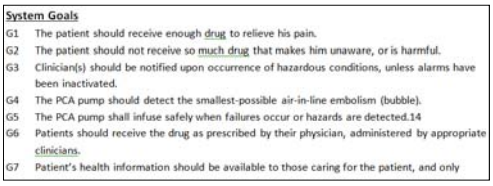
Process/Artifacts *(Steps 1-3)*



Project Glossary

Basal	Base rate of drug infusion
Bolus	Single dose of a drug or other medicinal preparation given all at once
Btty	Battery
CCBClinician-requested bolus.	
and so on…	

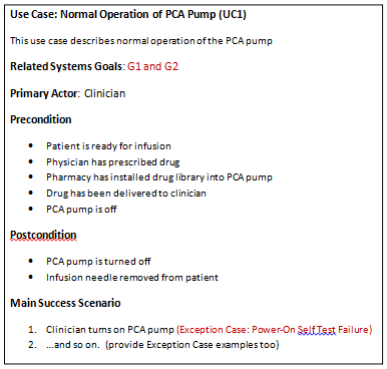
Step 1: Start creating a glossary of terms. Add to it incrementally *(Doors Next)*



System Goals

G1	The patient should receive enough drug to relieve his pain.
G2	The patient should not receive so much drug that makes him unaware, or is harmful.
G3	Clinician(s) should be notified upon occurrence of hazardous conditions, unless alarms have been inactivated.
G4	The PCA pump should detect the smallest-possible air-in-line embolism (bubble).
G5	The PCA pump shall infuse safely when failures occur or hazards are detected.14
G6	Patients should receive the drug as prescribed by their physician, administered by appropriate clinicians.
G7	Patient's health information should be available to those caring for the patient, and only

Step 2: Define clear system goals *(Doors Next)*



Use Case: Normal Operation of PCA Pump (UC1)

This use case describes normal operation of the PCA pump

Related Systems Goals: G1 and G2

Primary Actor: Clinician

Precondition

- Patient is ready for infusion
- Physician has prescribed drug
- Pharmacy has installed drug library into PCA pump
- Drug has been delivered to clinician
- PCA pump is off

Postcondition

- PCA pump is turned off
- Infusion needle removed from patient

Main Success Scenario

1. Clinician turns on PCA pump (Exception Case: Power-On Self Test Failure)
2. …and so on. (provide Exception Case examples too)

Step 3: Create 3-5 high-level use cases. *(Doors Next)*

Outcome: You know what your product should do.

22

Process/Artifacts *(Steps 4-)*

Coding starts

Step 4: Identify technical risks and plan/start work on architectural spikes.

- Interface with physical devices
- Mobile iOS/Android
- Web-Service
- Project environment
- Tools
- Team collaboration mechanisms
- Other project specific??

Step 5: Start defining requirements or user stories (agile/upfront?) *(Doors Next)*

Step 6: Start defining performance/quality related requirements.

Step 7: Safety Analysis. Construct a FMECA *(Doors Next)*

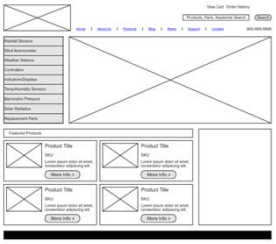
Item	Potential Failure Mode	Potential Cause of Failure	Current Prevention Controls	Current Detection Controls	Recommended Action
Disk Brake System	Vehicle does not stop	Mechanical linkage break due to corrosion	Designed per material standard MS-845	Environmental stress test 03-9963	Change material to stainless steel
		Master cylinder vacuum lock	Carry-over design with same duty cycle requirements	Pressure variability testing on system level	None
		Loss of hydraulic fluid due to back off of connector	Designed per torque requirements - 3993	Vibration step-stress test 18-1950	Modify connector from crimp style to quick connect.
		Loss of hydraulic fluid due to hydraulic lines crimped or compressed	Designed per material standard MS-1178	DOE tube resiliency test	Modify design from MS-1178 to MS-2025 to increase strength.

Outcome: You know more about your product.

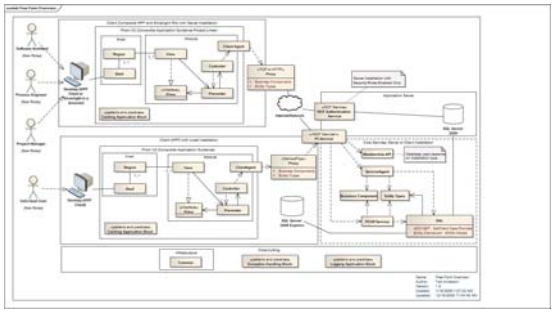
23

Process/Artifacts *(Steps 4-)*

Step 8: Security Analysis Security Cards or EOP *(Doors Next)*



Step 9: Architectural Design. Consider multiple solutions. Use UML.



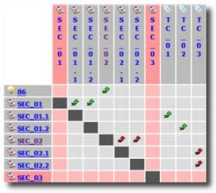
Step 10: UX Design
Mockup your main screens.
Determine task flows.

Outcome: You know that your product is viable.

24

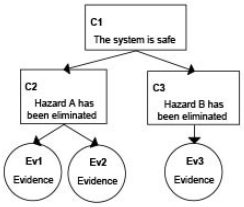
Process/Artifacts *(Steps 4-)*

Step 11: Code, Test (Start this early – for architectural spikes)



Steps 1-12: Create trace links as-you-go.

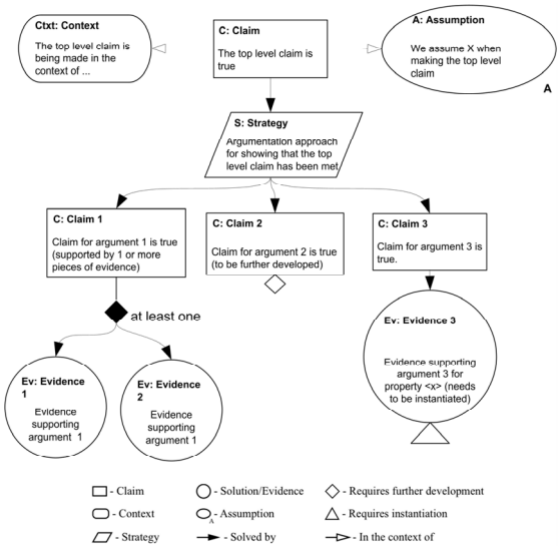
Step 12: Build an initial safety-assurance case. *(Mona will help)*



Outcome: You deliver a fully functioning architectural spike.

25

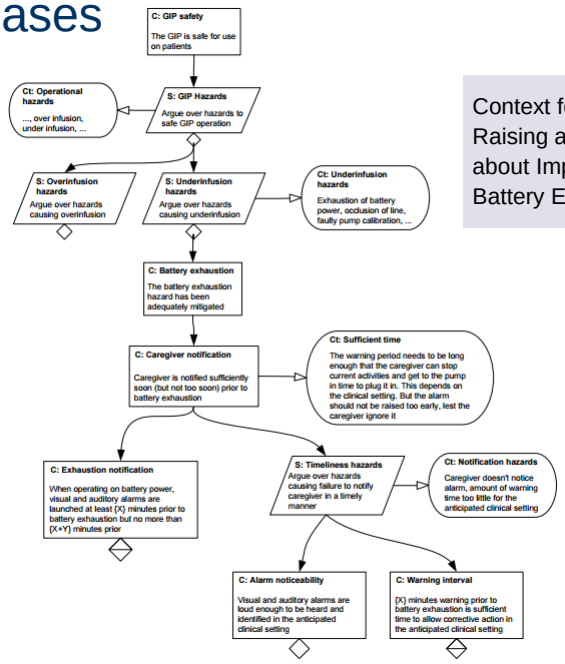
Safety Cases



Example GSN Argument

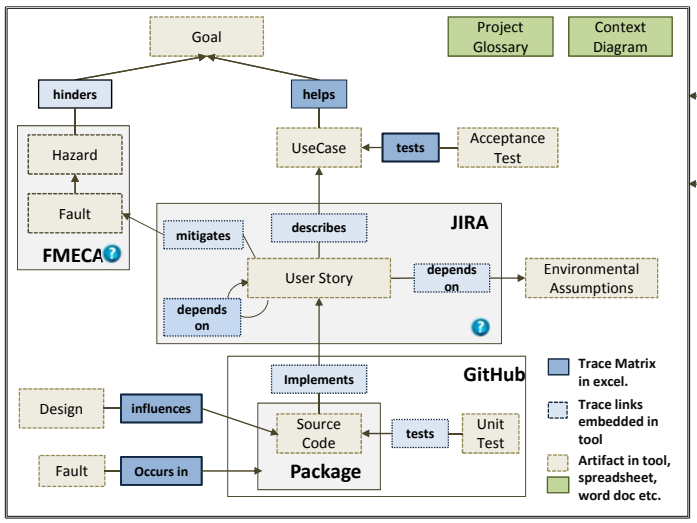
26

Safety Cases



Context for Raising an Alarm about Impending Battery Exhaustion

Managing the Artifacts



Safety Assurance Case
—Compiles evidence for safety

Threat Model
—Documents security vulnerabilities

This is an interactive Traceability Information Model.

Click on any node to modify/view data.

Commit your changes via Github.

28

SE491 vs. 591 Deliverables

SE491	SE591
• Approved executable architectural spike. **Version 0.01** • UI defined for mobile, web-server, and other interfaces. • All major risks mitigated (or with well-defined mitigation plan). • FMECA, Safety-Case, and all trace links in place for v0.01.	• **Fully functioning software system deployed**. Two deliverables for weeks 5 and 10. **v0.02**, **v1.0**. Each deliverable must have updated artifacts (including FMECA, Safety-Case, TraceLinks, Test Cases) • **Presentation to an industry panel** composed of UX Designer, Mobile Developer, and more.. • **Open field test** of fully functioning system.

See quarter schedule for specific deliverable/meeting dates.

29

Class Schedule (See D2L!!)

SE 491 Studio Schedule

Week	Date	Activity	Deliverables Due
1.	01/06/16	Overview of Studio Introduction to 5 projects Safety-Critical Development	01/09/16 Individual Surveys
2.	01/13/16	Tools and Environments Teams Formed	
3.	01/20/16	Team Meetings	
4.	01/27/16	Team Presentations (6.30pm)	**Presentation:** System Goals Project Glossary (initial) Use Cases (1-3) Initial FMEA GUI (mobile/services)
5.	02/03/16	Team Meetings	
6.	02/10/16	Team Meetings	**Check Point:** Architectural Design Review Requirements (EARS) Initial Safety-Case Review Code/Architectural Spikes Interface to Devices working
7.	02/17/16	Threat Modeling activity with Security Cards (Classroom 5.45pm-6.30pm, then group time for activity)	
8.	02/24/16	Team Meetings	*Dr. Huang out of town at IFIP Meeting*
9.	03/02/16	Team Meetings	
10.	03/09/16	Team Presentations (6.30pm)	**Presentation:** 20 minute final "marketing style" presentation.
11.	03/16/16	SE491 Final Submission	**Final Submission:** A report that describes work accomplished, collates artifacts from multiple tools, and includes a risk assessment and a prioritized list of

* Teams will meet with Dr. Huang at a scheduled time during Team Meetings. Additional teams may meet with Robert Cole (Device Expert) and/or Mona Rahimi (Safety Expert)

Limited lecture time.

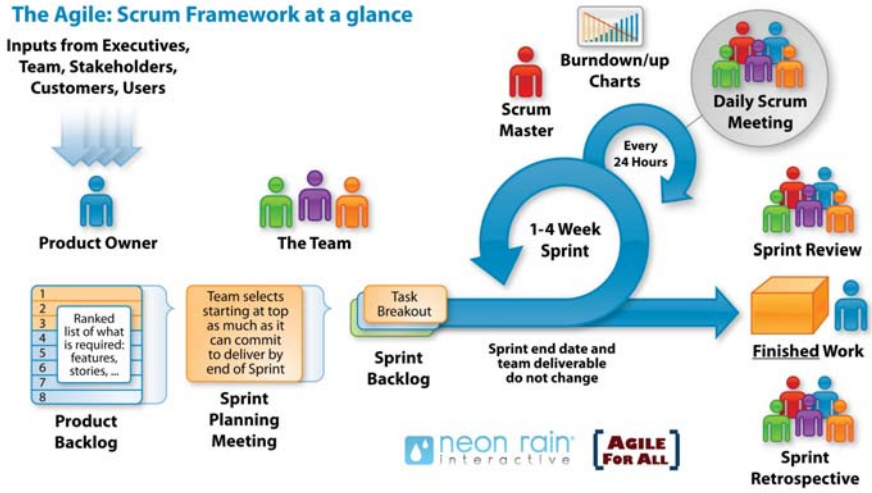
Mainly team work.

Everyone plays a role to deliver safe, working software!

Schedule is on D2L.

30

Weekly SCRUM meetings



Treat each weekly meeting as a SCRUM meeting. This will work best if you appoint a SCRUM master. You can switch roles every few weeks if you wish.

31

Team Work

https://www.youtube.com/watch?v=fUXdrl9ch_Q

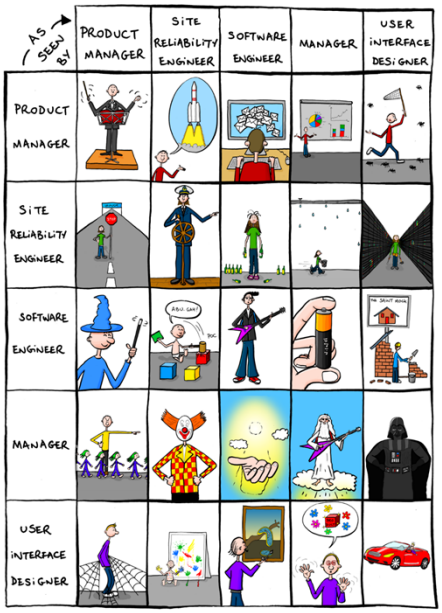
- Each project is non-trivial – so you will need to work closely with your team.
- People will have different skill levels. This is inevitable. Whatever your skill level, work hard, put in effort, make sure you review and understand your team-mates' work.

- **Don't** be the lazy team member!
- There will be peer-review. This tends to catch the most egregious cases.
- There will be a team-blog for documenting your contribution each week.

32

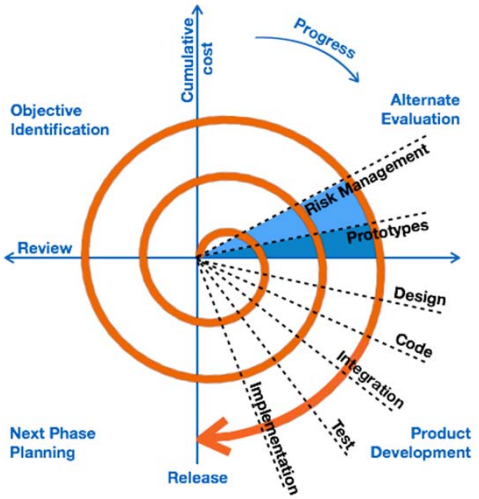
Studio Roles

- While there will be plenty of opportunities for cross-role experience, Studio will work best if each person is accountable for a specific task.
- The team is responsible for the project getting completed.
- Tasks can be shared by multiple team members (consider pairing experience levels).
- Responsibilities and tasks can (and will) be reassigned as the project progresses.



33

Think Iteratively



Barry Boehm's Risk Driven development is all about planning iterations that mitigate current project risks.

34

Suggestions for Initial Roles

Everyone writes Requirements!
Everyone Codes!
Everyone tracks progress!

- **Team Lead*:**
  Choose this role wisely – based on experience, people skills, and time availability. Scrum Master. (This person needs to assume additional technical role.)
- **Artifact Manager*:**
  Organize Jira Attlassian and GitHub repository
  Track artifacts (goals, use cases, requirements, trace matrices)
- **Architect/Developer***:
  Mastermind the overall architectural design. In early phases of the project setup infrastructure. Identify services (e.g. Google Maps). Run architectural spikes, develop executable prototypes.
- **Physical Device Expert***:
  Expert in physical devices of your project. Learn the interfaces. Write simple code to connect with devices. Create prototypes.

35

Suggestions for Initial Roles

Everyone writes Requirements!
Everyone Codes!
Everyone Tests!

- **User Interface Specialist*:**
  Mock-up UI design. Test UI design with initial user base.
- **Algorithm Specialist*:**
  Some projects (e.g. MedFleet and Aerial reconnaissance) requires quite complex algorithms and/or tool/framework dependencies. Identify such issues early and assign to team members.
- **QA**:
  Plan early how you are going to integrate QA activities into your lifecycle.
- **Presentation Coordinator**:
  Everyone on the team should participate in team presentations; however, it is often helpful to have a coordinator.

36

Grades

Group Grade 75%

- **Week 4**: Goals, Use Cases, Starting Glossary, FMECA, Exception Cases, (DOORS)
  GUI Design (Mockup) etc 20%
- **Week 4**: Presentation 5%
- **Week 6/7**: Checkpoint: Architectural Design, Requirements, Safety Case, Architectural Spike (20%)
- **Week 10**: Presentation 5%
- **Week 11**: Executable Architecture, Spikes etc showing all major development risks mitigated.
  Final Report (25%)

Individual Grade (25%)

5% Survey with questions

10% Practical Research report (video or tutorial)

10% Personal Contribution to team project as documented in weekly team journal, and as observed through regular meetings with Professor.

Most team submissions this quarter will be via selected tool environments – and as presentations. The exception is the final report.

** I reserve the right to assign lower grades to under-contributing team members.*

37

Project Environment

- **Atlassian Jira**:
  For managing issues, sprints, tracking features requests
- **Github:**
  Version control – interfaces with JIRA
- **BlueMix:**
  Hosting and integrated Android development.

38

Atlassian Jira:

Accounts will be created once teams are assigned!!

Jira is an issue tracking product developed by Atlassian. It provides bug tracking, issue tracking, and project management functions.

The name JIRA is a truncation of *Gojira*, the Japanese name for Godzilla – a reference to JIRA's main competitor, Bugzilla.

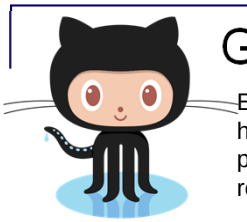
Live demo

You will receive an email invitation to join your JIRA Studio project shortly after groups have been assigned.

The **artifacts manager** will be given administration privileges for your project.

- Our Atlassian server http://sarec.cstcis.cti.depaul.edu:8090/secure/Dashboard.jspa
- Getting started guide https://confluence.atlassian.com/jirasoftwareserver070/getting-started-with-jira-software-762877200.html
- Jira in a nutshell (3 minutes) https://www.youtube.com/watch?v=xrCJv0fTyR8
- Online Tutorial (YouTube 40 minutes) https://www.youtube.com/watch?v=NrHpXvDXVrw

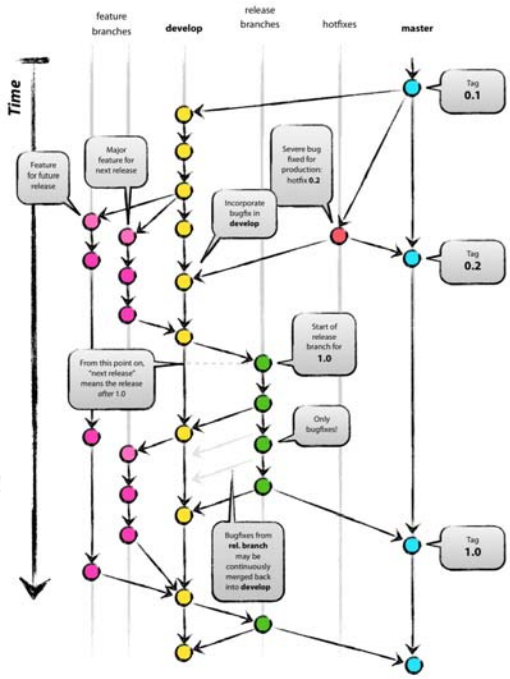
39



Github:

Each team has their own private repository.

- Projects will be made public at the end of SE591 under an MIT license. Students may opt to use pseudonyms if they wish. (View License in GitHub)
- All teams will follow the branching strategy on the right unless somebody has a better plan – but needs to be discussed in WEEK 1!!
- All artifacts will be committed to SVN at time of release.
- Three planned formal releases. End of SE491 and twice during SE591.



BlueMix:

IBM Bluemix is a cloud platform as a service (PaaS) developed by **IBM**. It supports several programming languages and services as well as integrated DevOps to build, run, deploy and manage applications on the cloud. **Bluemix** is based on Cloud Foundry open technology and runs on SoftLayer infrastructure.

We have free academic accounts for each student on BlueMix.
Learn more about it here:
http://www.ibm.com/developerworks/cloud/library/cl-bluemixfoundry/

41