02. Project Discussion

SPORTS TECHNOLOGY

Prof. Priya Narasimhan
Electrical & Computer Engineering Department
Carnegie Mellon University

Electrical & Computer ENGINEERING

Lecture material and readings are posted at
http://www.ece.cmu.edu/~ece549

Overview of Lecture

- Some project ideas
  - Sponsored projects
- Project outline
  - Expectations
  - Milestones
- Team website (the only documentation you need to provide for your project)
- Upcoming presentation guidelines
- Project status check

Project Ideas (1)

- Yellow line on cellphone
- Augmented reality
  - Examples: For the New York subway system
  - Basically, you are augmenting reality (real, live video) with other sources of information to heighten the experience
- Yellow (1st down) line in football
  - Indicates where the next 1st down line should be
  - Indicates how many more yards need to be made in the current play
- What if you could overlay the yellow line on top of the field, as you see it through the video camera of your cellphone?
  - Target sport: Football; Target user: Fan in the stands
- Requires you to know
  - Current stats – where is the 1st down line ?
  - Orientation and location of the field
  - Recognition of where to overlay the 1st down line
- This is a sponsored project

3

Project Ideas (2)

- Advertising analytics/impressions
- Impressions count
  - Advertisers and sponsors want to know logo-hit counts
  - Impression count or click count
- How does an advertiser determine impression count?
- Manual counting of how many times a logo appears
- Automated impressions engine
  - Target sport: Any (but we can obtain hockey footage)
  - Target user: Sports team, sports marketer/sponsor
- What if you could analyze raw video footage
  - Provide the number of times a logo was visible 100%
  - Provide the number of times a logo was visible 25%, 60%, etc.
  - For example: Reebok visible 100% in the picture above, and UPMC Sports Medicine visible only 60%
- This is a sponsored project

4

Project Ideas (3)

- Play Coach Tomlin
- What if you could "predict" what Coach Tomlin was going to call, for a specific play?
- Imagine if you are sitting in the stands and you can key in what kind of play Coach Tomlin was going to execute
- You can be in one of two player roles
  - Predictor: Predicts what the next play will be
  - Assessor: Assesses what the play was, after the fact
- Tally up the two sets of votes and assign points to people
- Assessor gets points if that assessor's play breakdown was the most likely
- Predictor gets points if that predictor's play prediction was the most likely
- Player can be in one or the other role for a given play, but not both
- This is a sponsored project

5

Project Ideas (4)

- Concept – electronic body suit for gaming
  - Wireless gaming experience using your entire body
  - Transmitting devices on a body-worn "suit" monitor the actions of the gamer
  - Static embedded nodes installed in the room communicate with the electronic body suit to record the user's actions
  - Information from the static nodes works with a program on a larger footprint receiving device (e.g., a PC, smart phone) to orchestrate various games
  - Game status might be displayed on a graphical interface (on a PC or PDA)
- Target sport: Video games; Target user: Players
- What's involved from a technical standpoint
  - Use of motion, tilt and vibration sensors
  - Communication between the body suit and the static nodes
  - Localization algorithm
    - Example: RSSI – Received Signal Strength Indication
  - Timing-critical programming to ensure "live" display of game

6

Project Ideas (5)

- Accelerometer-based sensing/training of players
- Given accelerometer data, what can we tell about players or the way a game is played?
- Can we build portable accelerometer devices (combined with other sensors) that could tell a player whether he/she is performing better?
- Target sport: Hockey (for instance); Target user: Player, coach
- What if you had accelerometers on the hockey stick, the player's pads, and the player's helmet?
- How does that data correlate with the sports and the statistics?
- Do players that shoot faster get the most goals?
- Do players whose feet move in tune with the stick get more saves?
- What is the relative acceleration of players in a shift?
- You would need to work with the CMU hockey team (or others) to get these answers

7

Project Ideas (6)

- Fantasy football (or any other sport) integrated with set-top box
- Ever watched a game
  - Trying to catch the game live to TV?
  - Trying to check your fantasy football stats on your favorite website (Yahoo, for example)?
- What if you had an integrated experience?
  - Customized docking of your specific fantasy game stats shown as a running ticker during the broadcast
  - The fantasy football stats should be updated in real-time (with alerts), customized to the person watching it
- Target sport: Fantasy football (or your favorite fantasy league); user: fan or the fantasy-football addict
- This is a sponsored project

8

Emphasis on Sports Technology

- Understand the target sport
- The target user: One or more of the following
  - Coach, player, sports marketing personnel, scout, General Manager, sports doctor, trainer, fan in the stands, fan at home, broadcaster
- Game's regulations and constraints
  - What are you not allowed to do, according to the game's regulations?
  - What are you forbidden from sensing/viewing/measuring?
- How will you test your "product"?
  - Need to ensure that you have your test demographic lined up
  - We will file IRB documents, as needed
  - You will need to present data to show that you've validated your product
- Pricing and requirements
  - What is the ideal price-point for your "product"?
- Interface
  - What is the interface to your target user(s)?

9

Project Phases & Expectations

- Concept
  - Need, competitive analysis, components, procurement
  - Target sports, target user
- Requirements
  - Functional: Logical features for prototype to work
  - Non-functional (quality of service): Reliability, security, timing, performance
- Specification
  - Design, architecture, hardware, software, interfaces
  - User interactions, subsystem interactions, use cases
- Prototyping – iterative
  - Skeleton functionality, quality of service, user interface
- Quality assurance (testing, empirical evaluation)
  - Test cases, unit test, integration test
  - Measurement of quality of service (as defined by requirements)
- Packaging and demonstration
  - Usability, aesthetics, demonstration script

10

Project Documentation

- Every team will be assigned a team directory and a team number (in the order in which I received your original project proposal)
  - /afs/ece/class/ece848d/fall09/teamX
  - Only that team's members have write access to that AFS directory
  - Team's project website to appear as http://www.ece.cmu.edu/~ece848d/fall09/teamX
  - Should be available by the end of the day
- HTML documentation template for project website (end of the day)
  - Available at http://www.ece.cmu.edu/~ece848d/resources.html
  - Copy the template over into the root of your team directory as index.html and edit
  - All of your milestones should be submitted through your team's project website
  - Feel free to embellish and be creative – this template is just a starting point containing items that I would like to see on your project website
  - We will walk through the outline today
    - Using the Trinetra system (http://www.ece.cmu.edu/~trinetra) as a running example

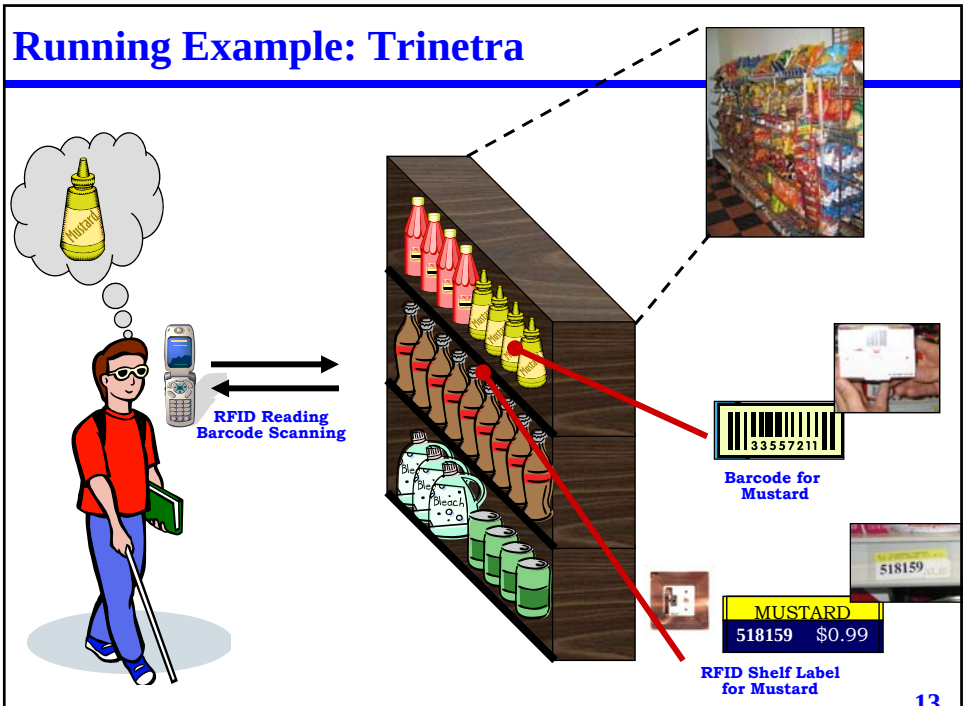
11

Documentation Template for Project Website

- Has subsections corresponding to the project phases (see slide 3)
- Needs to be completed for each project milestone
  - I will consult this before your team shows up for a meeting in my office
  - I will look for updates after each milestone passes
- Please keep your team's website up to date with
  - Design, architecture and test documents
  - Working versions of code
  - Presentations made in class
- Feel free to add your own fun touches
  - Telling the "story behind the story" – what makes your project great
  - Pictures of your prototype in progress through the phases
  - Pictures of your team members hard at work, goofing off or sleeping ☺

12

Running Example: Trinetra



13

Project Website: Concept

- Fill this section out before your 1st in-class team presentation
- 3 sentences that represent an "elevator pitch" for your project
  - 1st sentence – description that is accessible to the layman that is more or less free of technical terminology, but conveys the impact of your project
  - 2nd sentence – more technical description that is still accessible to the layman, but that articulates the way in which your project accomplishes its goals
  - 3rd sentence – which sport are you targeting and which demographic (user) are you targeting

- Example
  - 1st sentence – This project will allow a blind person to enjoy a greater degree of independence in grocery shopping in a cost-effective, portable way.
  - 2nd sentence – The prototype integrates off-the-shelf embedded components such as a Bluetooth-enabled smart phone, text-to-speech software and a scanner capable of reading barcodes off grocery products in real time.
    - 3rd sentence – Does not apply here

14

Project Website: Motivation

- Fill this section out before your 1st in-class team presentation
- 2-3 sentences that represent the need for your project
- Often also viewed as the "market differentiator" for your project
- What does your project fundamentally make possible that is
  - Not available today?
  - Far too expensive today?
  - Difficult to use today?
  - This is the brand-new/faster/better/cheaper argument
  - 1st sentence – how are things done today that is inadequate in some way that drives the need for your project? (the "before" scenario)
  - 2nd sentence – how will things be different after your project? (the "after" scenario)

- Example
  - 1st sentence – A blind person today requires assistance from a sighted individual (a friend or a store clerk) in order to purchase grocery items, thereby hampering his/her independence.
  - 2nd sentence – Using our prototype system, a blind person will be able to navigate a grocery store, free to locate, identify and purchase items on his/her own, unassisted.

15

Project Website: Competitive Analysis

- Series of bullets, each describing a product or feature that competes with yours
- These can be off-the-shelf products or academic projects
- For each bullet, provide
  - Image of the product in use
  - Name of the product, vendor, pricing, with link to URL
  - 1 sentence about the competing product or project
  - 1 sentence about how your project differs from this competitor
    - Differentiators: Cost, portability, features, …..

- You can have as many bullets as you like
  - Not acceptable to have zero bullets or just one – this would mean that you did not do your competitive analysis correctly

- Caution
  - Do not be disparaging of others' work
    - Avoid sentences like "This product is useless." or "This product is poorly documented"
    - Stay factual and objective – this is what you would do in industry

16

Project Website: Requirements

- Series of bullets, each describing a key non-functional characteristic of your system
- Functional requirements (4-5 bullets at least)
  - Logical characteristics or operations of your system
- Non-functional characteristics include
  - Reliability, safety, security, portability, timeliness, performance, durability
- For each bullet, state the requirement that you expect/want your project to meet
  - Clearly, we might not reach that goal – that's okay as long as we understand why and what the new reality of the requirement is
- Example
  - Functional requirement: Accepts barcodes as input information, accepts RFIDs as input information, converts barcode to descriptive product information to display on phone, converts phone display to audio input to the user, etc.
  - Timing requirement: Provides a response to the blind end-user within no more than 5 seconds of the user scanning a barcode
  - Reliability requirement: Continues to provide service even under the loss of a single message. The Trinetra system will not protect against hardware crashes.

17

Project Website: Technical Specifications

- All of the components used to build your prototype
- Think of this section as the "technical specs" section of your product, if someone were to purchase it
  - Have three subsections – hardware, software, protocols
- List all your hardware components
  - For each component, provide technical specifications (vendor, processor speed, memory, storage, etc.) – does not hurt to be detailed
  - For each component, provide a link to the vendor that we will purchase this from (if that is the case)
  - For each component, provide an estimated cost, if you have the number
- List all your software components
  - For each component, provide technical specifications (vendor, programming language, compliance to standards) – does not hurt to be detailed
  - For each component, provide a link to where the software can be downloaded
- List all protocols used

18

Project Website: Components

Example picture shown here for the Trinetra system
We will expect a similar picture for your system

19

Project Proposal on Team Website

Team Members
- insert names of team members

Concept
- insert material requested on slide #7 of this lecture

Motivation
- insert material requested on slide #8 of this lecture

Competitive Analysis
- insert material requested on slide #9 of this lecture

Requirements
- material requested on slide #10 of this lecture

Technical Specifications
- insert material requested on slide #11 of this lecture
- insert pictures similar to those shown in slide #12 of this lecture (clipart and web images will do for now – once you get the real components, insert a real photograph)

Due on 9/2 before you come to your team presentation

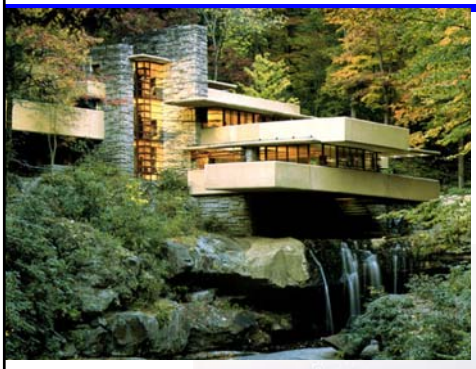
20

Project Design & Architecture

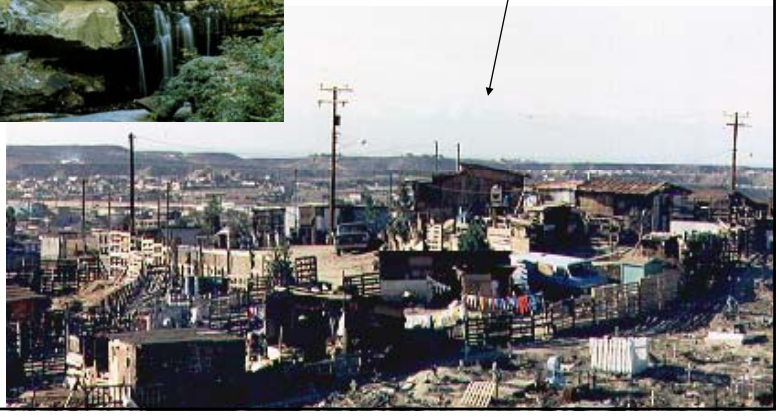
- What does architecture mean here?
- Blueprint of the design of your system
- Includes
  - Components involved in your system and how they are related
  - Use cases that show the various scenarios that matter in your system
  - Interaction diagrams that show the use of the various components
  - Physical deployment scenario

21

Good vs. Bad Architecture



You get to decide which of these your project will resemble the most!



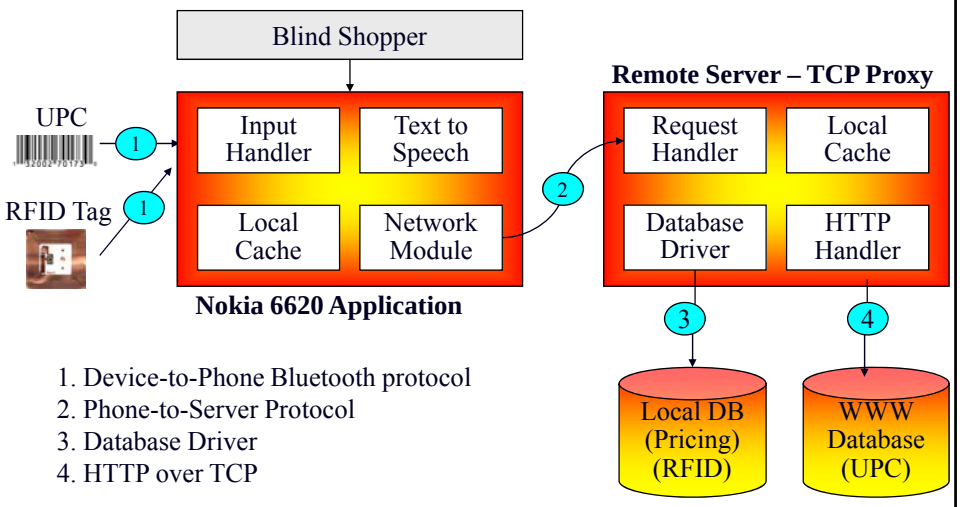
From Requirements To Architecture

- Example: Sample functional requirements (not a complete list)
  - "Accepts barcodes as input information"
  - "Accepts RFIDs as input information"
  - "Converts barcode to descriptive product information to display on phone"
  - "Converts phone display to audio input to the user"
- What does this translate to, in terms of the application?

- Look at the kinds of components you need
  - What is each component's function? (What are the methods of the interface?)
  - Who does it invoke? Who invokes it? Through what protocol?
- Look at the kinds of exceptions/errors generated
  - What kinds of errors can arise at the component?
  - How should each component handle an exception that it sees?
- End-user interfacing
  - How many end-users can be supported?
  - What kind of input and output to the end-user?

23

Strawman Architecture



1. Device-to-Phone Bluetooth protocol
2. Phone-to-Server Protocol
3. Database Driver
4. HTTP over TCP

24

Oops! No Behavioral Specification!

25

Why Specify Behavior?

- Isn't it enough to specify interfaces?
  - No! Why?
  - Would you buy a car without taking a test-drive?
- Behavior (could also be component-level or system-level)
  - What should the object do?
  - What should the object not do?
  - How does the object respond to external invocations/stimuli?
  - How does the object meet each of its requirements?
  - What are the state transitions that the object undergoes?
  - What is the object's response to a fault? (reliability behavior)
  - What is the object's response to overload? (performance behavior)
  - What is the object's response to a missed deadline? (timing behavior)

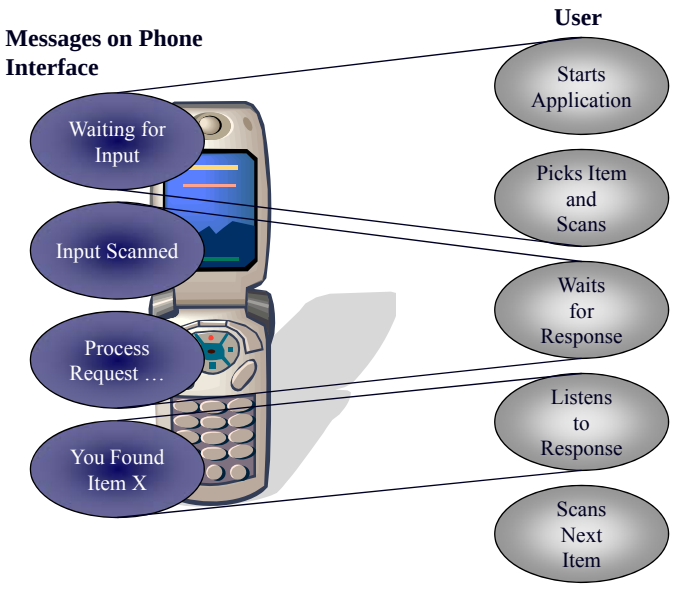
26

Interaction Diagrams (or Use Cases)

- Think about what you're promising to deliver to us
  - Not just what the interfaces look like, but how the interfaces behave
- Enumerate the behavior of each interface under
  - Normal messages from, and to, users of your system
  - Normal messages from, and to, the database
  - Abnormal messages from, and to, your users
  - Abnormal messages from, and to, the database
  - Abnormal messages from third-party products
  - Environmental conditions (e.g., the march of time)
  - Faults that crash machines or objects
  - Recovery of failed machines or objects
  - Conditions that might result in missed deadlines
  - Conditions of overload
- These are called interaction diagrams or use cases

27

Example User-Interaction Diagram



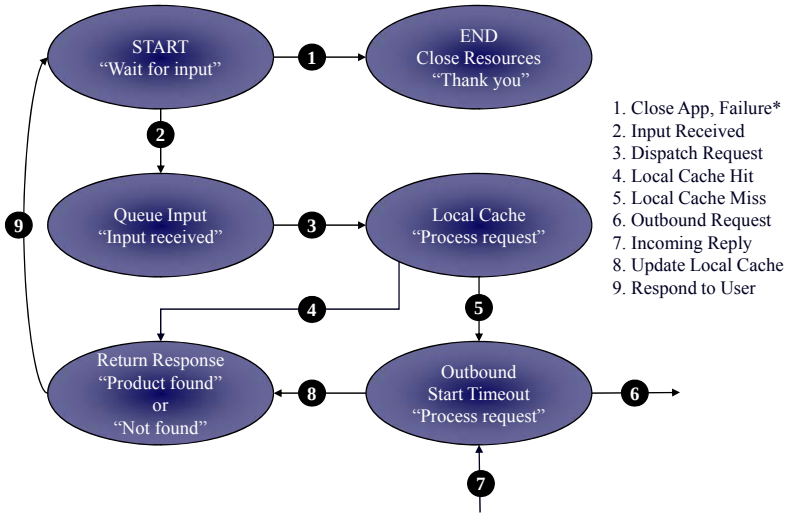
28

Affairs of State

- State is often the most important thing in several system issues
  - Functionality
  - Reliability
  - Real-time
  - Performance
- Ask yourself key questions when designing your interfaces/implementations
  - What states can your system find itself in?
  - What variables change within each state? And as state transitions occur?
  - How often do state transitions occur? Is there a limit?
  - How often is this state being updated? Which messages trigger state transitions?
  - What if something bad (e.g., a fault) happens?
  - If I have many users (overload), how is this indicated?
- Think in terms of (relatively simple) state-transition diagrams, where messages (within your system) form the triggers for state transitions

29

Example State Transitions Within System



30

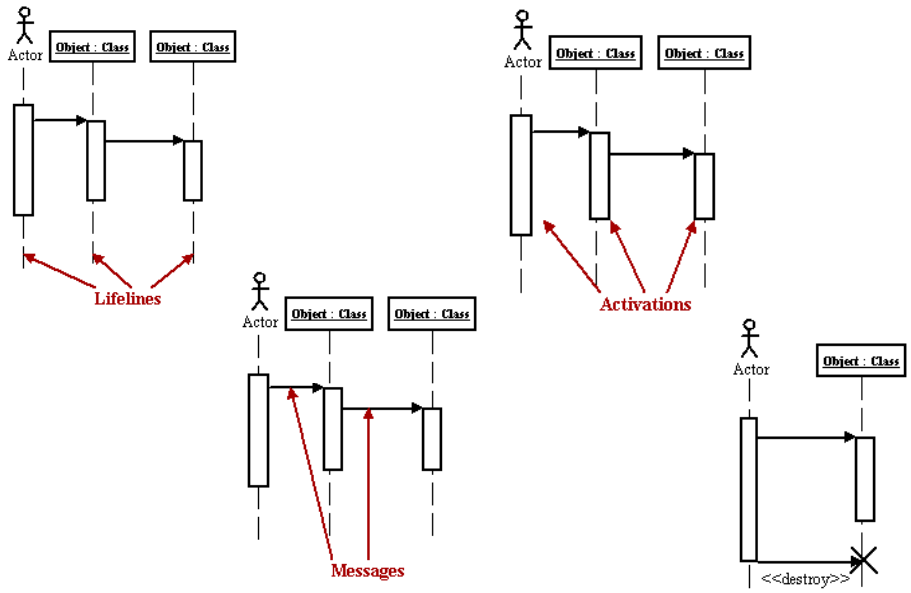
Use Cases ⇨ Sequence Diagrams

- We are using a lot of UML concepts in a lightweight way
- Sequence diagrams – way of representing a use case through interactions across "objects" in your system and the messages that they exchange

- Vertical lines represent *objects* in your system
  - Label each vertical line with a box on top indicating the object's name
  - The vertical line is also called the *lifeline* – an X on this line indicates that the object has died
- Horizontal lines represent *messages* in your system
  - ⟶ represents synchronous message (message sender "blocks" until the operation completes)
  - ⟵ represents an asynchronous message ("fire-and-forget" message where sender does not "block" for the operation to complete"
  - ⟵ - - - represents a return of a message. Label arrow with the name of the returned data structure – unnecessary for methods that don't return values
- Vertical boxes represent *activations* – the amount of time that the object needs to complete a task

31

Simple Examples



32

What Does Traceability Mean?

- Traceability = ability to define and track relationships between entities
- What entities are we interested in tracking?
  - Requirements
  - Architecture
  - Scenarios or use cases
- Why do we want/need traceability?
  - Helps you and us verify that requirements have been met
  - Helps your team develop acceptance tests as you go along
  - Helps your team to manage change/upgrades easier
- Yes, there is a cost (overhead of maintaining traces), but the payoff is huge – your systems will be easier to maintain, change and test!

33

Interfaces From A Traceability Perspective

- Your team probably understands your interfaces very well
  - How about someone outside your team who does a design review?
  - How about the instructor and the TAs who need to know if you have achieved your requirements?
- Need a common way of expressing information inside every interface or class so that it's understandable and traceable
- Need to do this within your source code as well
- This is what we will look for when we do a design review of your team's interfaces

34

Freezing Your Code

- What's a "code freeze"?
  - Stable version of code that works
  - Complete distribution that your team declares to be bug-free
  - Has a version number (e.g., 1.0)
  - Frozen code cannot be modified (changes and bug-fixes go into the next higher version)
- Why should you freeze your code?
  - Working version of *some*thing (not *every*thing) that you can go back to
  - We can test that version

35

Team Responsibilities

- Responsibilities within each team to be divided amongst members
  - Each team member implements part of the team's software
  - In addition, there are key project management roles to consider
- Key project management responsibilities
  - Hardware, software, implementation – all team members
  - User interface
    - Defines the interface to the user and the associated requirements
  - Testing + test documentation
    - Writes test scripts and generates documentation for tests
  - Build + webmaster + external documentation
    - Hands the code off to instructors and keeps the website updated
  - Workloads
    - Responsible for developing synthetic workloads
- Appoint one person for each of these responsibilities
  - I will definitely want to know this information at a future design review

36

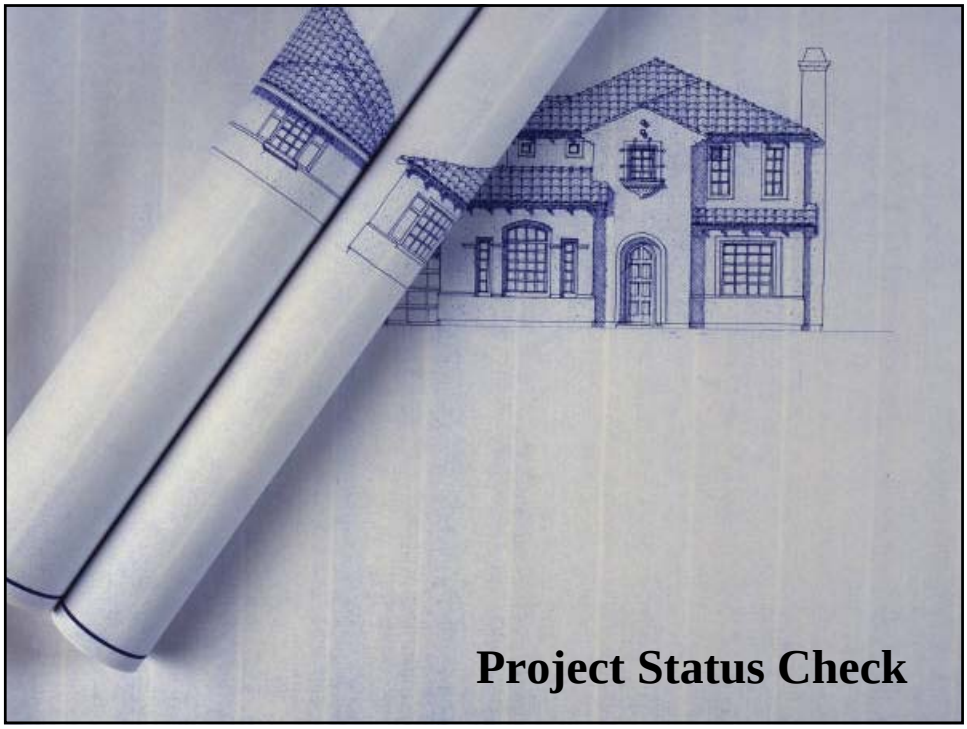
Project “Scapegoat” Flowchart

37

What Are Your Challenges?

- ✓ Build up your sub-project in stages
  - ► Minimal working implementation first (one end-to-end use case working)
  - ► Dependability architecture
  - ► Real-time architecture
  - ► Performance architecture
  - ► User interfaces
- ✓ Plan for and build in testing hooks
  - ► Design & write acceptance tests
  - ► Do not forget exception-handling
- ✗ Resist the urge to
  - ► Add “bells and whistles” first
  - ► Start coding without a design
- ✓ Break up your sub-project and distribute pieces to different team-mates
  - ► Each person to have a specific role/responsibility within the sub-project
  - ► Team work is often challenging, but a necessary skill to cultivate
- ✓ Be imaginative, be creative, go beyond the mundane!

38



What You Should Be Doing Now

- Put together a "cool name" for your prototype – we will use this name to refer to your project henceforth
- Reach agreement on the project idea that your team will pursue
- Actively research the procurement of parts for each of your project ideas
  - Evaluate how long it will take to order them, and your budget
  - Do not forget all the cautionary advice on choosing unsupported hardware
    - See first lecture
- Update your team's project website before September 2nd with
  - Concept information, motivation, competitive analysis, requirements and technical specifications
- Get ready for your first in-class presentation on September 2nd
  - Outline follows

40

Project Team Website – Expectations

- What's due imminently
  - Due on Sept 2nd – class presentation
  - Concept section
  - Competitive analysis section
  - Motivation section
  - Requirements section
  - Components (technical specifications) section
  - Link to your class presentation for 9/2
- What is due next
  - Due on Sept 21st – project checkpoint
  - Class presentation to review progress
  - Architecture, use cases, interaction diagrams
  - Visit risks and mitigation strategies
  - Link to your class presentation for 9/21
- What is due after
  - Due on Oct 14th – mid-semester demo
  - Working prototype
  - Poster
  - Link to your poster for 10/14

41

Presentation Guideline

All teams present
Project Proposal
on 9/2

42

Outline of Presentation

- Maximum of 4 slides per team, excluding title slide
- Maximum of 5 minutes per team
  - Ideally, 1 minute per slide so that you only need about 4-5 minutes altogether to present
  - Remainder of the time for questions from the audience
  - You are graded on content and time management as well
- Only one team member does the entire presentation
  - Team members will take turns – next team presentation by a different person
- Upload your team's presentation to your team's website before class that day

- Title slide
  - Cool name for your team, team members
- Slide 1
  - Target sport, target user, concept, motivation (Before and After scenarios)
- Slide 2
  - Competitive analysis
- Slide 3
  - Requirements – to the best of your ability at this stage
- Slide 4
  - Technical specifications – parts that you need

43

Guidelines

- Think of a field-demo scenario that you will be able to show me
  - If you wanted to showcase your project, what would you put into a five-minute video clip to pitch your solution?

- Extras
  - Use pictures where possible
  - Pictures of parts, pictures of concept (Before/After), competing products, etc.
  - Team pictures also welcome ☺

- Success criteria
  - The class should be able to infer this from your presentation
  - How will you (and I) know that you're done?
  - Qualitative – "We will demonstrate feature X in our prototype"
  - Quantitative – "We will demonstrate that we can do X in *y* seconds for *z* users"

44

Review of Lecture

- Project outline
  - Expectations
  - Milestones
- Team website (the only documentation you need)
- Upcoming presentation guidelines

- First meeting with me
  - Monday, 8/31 – we will go in order, to be posted on the class website
- First in-class presentation by all the teams
  - Wednesday, 9/2 – we will go in order, to be posted on the class website

- Project status
  - What's done
    - Nothing so far
  - What's next
    - Meeting with me next week, finalizing your project ideas, documentation process started through team website, preparation for presentation on 9/2

45

Project Concept Cheat-Sheet

- Target sport:
- Target user(s):
- Aspect of the sport:
- Concept (2-3 bullets):

- Competition (pictures, prices, links, where used)

- Superiority of your product:
- Technology platforms (2-3 bullets)

46