PROJECT SELECTION

CS/ECE 3992

SOME SLIDES BASED ON AL DAVIS' 3992 SLIDES FROM SPRING 2013

Project Ideas

- No engineer looks at a television remote control without wondering what it would take to turn it into a stun gun. No engineer can take a shower without wondering if some sort of Teflon coating would make showering unnecessary. To the engineer, the world is a toy box full of suboptimized and feature-poor toys.
    - Scott Adams

Idea Selection

- Idea & Team == Chicken & Egg
  - The idea needs to be embraced by the team
  - The team skills need to fit the idea
- In the end the idea needs to:
  - Be fun, exciting, and something you'll be proud of
  - Have both an original SW and HW component

Features

- I don't care what it is, when it has an LCD screen, it makes it better.
  - Kevin Rose

Novelty?

- There isn’t really a novelty requirement
  - It’s OK to design something you could buy
    - Learning how things work can be a lot of fun
  - However, novelty does tend to add to the excitement
    - Demonstrating creativity is a good thing
- Show off your engineering chops!

Pragmatics

- Whatever your project is, it MUST be finished, documented and demonstrated
  - On time!
- Psychologically
  - If it’s fun, you’ll likely do it and do it well
  - If it’s drudgery it you and the project will suffer
  - You pick the project!
    - So, consider fun vs. drudgery…

Scope

- CS/ECE 4710 is a 3cr class
    - this implies 3hrs/week in class, and 6 hrs/week outside of class
- Initial scoping sanity check
    - 9 hrs of work * 15 weeks * # of team members
    - i.e. 135 hrs per person
    - This includes design, test, demonstration, documentation
    - Does NOT include planning, parts lead time, etc. (That's 3992 and summer)

Project Scope Planning

- Predict manpower estimates for all tasks
    - NOTE - few people can actually do this accurately
- Honest attempt to get this right will be a huge help in getting things finished in the Fall

Timetables

- Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law.

Scope Problems

- Things we often underestimate
  - How slow we are
  - Documentation time
  - Debugging / test time
  - Time lost due to screw-ups
  - Time lost due to people issues
    - Other classes, need to take a break, need to ski, etc.
    - Don’t plan on people being robots…

Deadlines

- I love deadlines. I like the whooshing sound they make as they fly by.
    - Douglas Adams

Scope Problems

- Another thing that is often underestimated
    - Group communication time
        - Regularly scheduled meetings are essential
            - Minimum is one meeting per week
            - Meeting minutes/results in a meeting log
        - Meetings can be short, but they should be regular

Group Scope = sum of components

- Each component gets assigned to an individual
  - group components are, of course, OK, but it's good to have a component lead
- Parallel efforts are the key to productivity
  - Only really works when interfaces are articulated, understood, and documented in advance
  - interfaces are complex – precise definition is required - don't wing it!
- Component-wise design, testing, and combination
  - Something always goes slower than you expect it to…

Setting Objectives

- Objectives are the specifics of what you will do
  - *Baseline set*
    - What you're sure you can pull off
    - Something you will still be proud of
      - This is really important!!!
  - *Wishlist*
    - What you hope you can also pull off if things go smoothly
    - And that will knock the socks off the judges

System Complexity

- A complex system that works is invariably found to have evolved from a simple system that worked.
  - John Gall

Optimization

- Premature optimization is the root of all evil in programming.
  - C.A.R. Hoare

Risk Management

- Every project has risks
  - people/parts/design/testing/whatever
- First step in managing risks: Articulate them!
  - Don’t go off the deep end - stay with reality
- Primary Plan: plausible avoidance of risk
- Mitigation Plan: what happens if Primary Plan fails?
  - How does project proceed without the risky component?
  - Finish project with equivalent or adequate substitute

SW/HW

- Hardware eventually fails. Software eventually works.
  - Michael Hartung

Surprises

- Every project has them!
  - Example: You pick a board or part that has lousy documentation
  - Example: You fry some part because you didn’t understand how to use it
  - Example: You order critical parts and they suddenly go on back-order for six months

Tool Choices

- Make sure you agree on tools that you know, or at least have some familiarity with
  - Programming languages, development systems, HW platforms, etc.
- Physical construction capabilities of the team
  - PCBs, enclosures, frames, physical platforms, etc.

Software

- I did say something along the lines of C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do, it blows your whole leg off.
    - Bjarne Stroustrup

Debugging

- Don’t underestimate debugging time in your planning
    - You can minimize debugging time through careful design and planning
    - You can minimize debugging time through excellent documentation at every stop of the project

Software

- Any fool can write code that a computer can understand. Good programmers write code that humans can understand.
    - Martin Fowler

Debugging

- Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.
    - Brian W. Kernighan

Software

- Always code as if the person who ends up maintaining your code will be a violent psychopath who knows where you live.
  - Damian Conway

Defining Success

- Define exactly how to know when objectives have been met
  - Should be articulated for the system as a whole, and for each component
- Demonstrate a capability
  - Requires defining a test and a way to score the result
  - This is what you’ll show on demo day!

Success and the Final Demo

- Critical because it influences your grade!
  - OK - that's just an operational point…
  - The main point: We're in a professional discipline
    - The best people get great jobs
    - The average people get less interesting jobs
  - This project can be great evidence of your skills and abilities to an employer!

Scheduling (Tricky!!!)

- Account for every aspect of the project
- Try organizing per-person and per-task GANT or PERT chart
  - basically a time line and dependence chart
  - includes resource requirements - parts, equipment, etc.

Scheduling (Tricky!!!)

- At any given point next Fall you should be able to answer:
  - what team member x is going to be doing on day y
    - This might be overkill - think of it as idealized target
  - Risk factors should be articulated clearly
  - Regular, meaningful milestones and test procedures should be clear

Time Management

- Managing your time without setting priorities is like shooting randomly and calling whatever you hit the target.
  - Peter Turla

Time Management

- Do not confuse motion and progress. A rocking horse keeps moving but does not make any progress.
    - Alfred A. Montapert

First Step

- Team selection and Idea Articulation
    - We’ll discuss project ideas as a group in class

Second Step

- Initial design flow
    - Interface design and specification
    - Initial design specification and schedule
    - Identify components
    - BOM (Bill of Materials)
        - You should plan on reading lots of data sheets!
        - Remember lead times for ordering parts
    - Risks
    - Proposal!
        - Detailed specification of all of the above

Specifications

- A specification, design, procedure, or test plan that will not fit on one page of 8.5-by-11 inch paper cannot be understood.
    - Mark Ardis

Initial Design → Approved Proposal

- Proposal contents (more later!)
  - Functional objectives
  - Top-level design
  - Tasking
  - Interface specifications
  - testing plan and process
  - module integration plan
  - risk analysis
  - schedule
  - BOM

User Interfaces

- I have always wished that my computer would be as easy to use as my telephone. My wish has come true. I no longer know how to use my telephone.
  - Bjarne Stroustrup

User Interfaces

- If your UI even vaguely resembles an airplane cockpit, you’re doing it wrong.
  - John Gruber

High-Level Design

- Creative part can be lots of fun
  - Remember - blue-sky needs to meet reality…
- HW and SW modules need to be specified
  - Be clear about what you’ll design vs. what you’ll acquire
  - Be clear about interfaces
- Everybody on the team must understand this high-level design thoroughly!

Design

- The dumbest mistake is viewing design as something you do at the end of the process to ‘tidy up’ the mess, as opposed to understanding it’s a ‘day one’ issue and part of everything.
  - Tom Peters

A Note on Help

- This project is about what your team can produce
  - You get to make the project choices for a change
- Feel free to learn from outside experts
  - Faculty , friends, colleagues, papers, books, etc.
  - Make sure sources are cited!
- BUT make sure that the actual design and implementation is done ONLY by the team!

Documentation

- 3992: Project proposal and presentation
  - Three phases: initial, intermediate, final
  - Iterative revision of the proposal/presentation
- 4710: Final project report
  - Thorough documentation of the entire project
  - Could someone else recreate it from your documentation?
- Continuous documentation throughout the project!!!

Documentation

- Most programmers regard documentation as necessary evil, written as an afterthought only because some bureaucrat requires it. They do not expect it to be useful. This is a self-fulfilling prophesy; documentation that has not been used before it is published, documentation that is not important to its author, will always be poor documentation.
  - David Parnas

Design

My engineering teachers laid down some basic rules:

- Design before implementing.
- Document your design.
- Review and analyze the documented design.
- Review implementation for consistency with the design.
  - David Parnas

Quotations

- The trouble with quotes on the Internet is that you can never tell if they are genuine.
  - Abraham Lincoln