

# Practical Project Management

CSCI 5040: Professional Master's Project (1 of 2)

Lecture 6

# Learning Objectives

- Review software project management practices
- Review weekly status reporting requirement
- Review design team activities

# Your experiences...

- Total failures, failed requirements, time overruns
- Late decision on change, new technologies, integrating technologies
- Vague requirements
- Unexpected issues, new technologies
- Lack of experience on team, lack of participation
- ...

# My experiences...

- Two large development projects (100 to 200 engineers)
  - 2 year project - 3 years late?
  - 2 year project – 3 months late?
- Project rescue (Bruce, go fix this)
  - Unhappy customers, vendors, management, developers
  - Mandatory overtime/no vacation
  - Project slip after project slip
- Fixing bad software or making it new?
  - History of poor software management and heroic deliveries
  - Late delivery, buggy software
  - Better delivery, limited scope, fix key issues
  - Move to new technology – leveraging web and cloud
- ...

# Classic view of Software Projects

- In 2000, per one study:
  - 23% of software projects failed
  - 49% were challenged (had difficulty meeting goals)
  - 28% succeeded
  - <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1438340>
- What were the main contributors to project failures?
- We'll start by looking at information gathered by Steve McConnell on classic mistakes and their reported frequency
  - <https://www.construx.com/resources/software-developments-classic-mistakes/>

# Classic Mistakes by Reported Frequency

- Overly optimistic schedules 77%
- Unrealistic expectations 73%
- Excessive multi-tasking 71%
- Shortchanged quality assurance 70%
- Noisy, crowded offices 69%
- Feature creep 69%
- Wishful thinking 68%
- Insufficient risk management 68%
- Confusing estimates with targets 65%
- Omitting necessary tasks from estimates 61%
- Abandoning planning under pressure 59%
- Shortchanged upstream activities 58%
- Heroics 58%
- Lack of user involvement 57%

# Classic Mistakes by Reported Frequency

- Inadequate design 54%
- Insufficient planning 54%
- Wasted time in the fuzzy front end 52%
- Planning to catch up later 51%
- Weak personnel 49%
- Undermined motivation 45%
- Unclear project vision 44%
- Requirements gold-plating 44%
- Code-like-hell programming 44%
- Lack of project sponsorship 42%
- Politics placed over substance 37%
- Adding people to a late project 36%
- Friction between dev & customers 36%
- Developer gold-plating 35%
- Lack of stakeholder buy-in 33%
- Trusting the map more than the terrain 32%
- Assuming global development has little impact 30%
- Outsourcing to reduce cost 29%
- Uncontrolled problem employees 29%
- Letting a team go dark 28%
- Subcontractor failure 27%
- Silver-bullet syndrome 26%
- Push me, pull me negotiation 26%
- Overestimating savings from tools/methods 24%
- Premature or too frequent convergence 24%
- Research-oriented development 19%
- Lack of automated source control 14%
- Switching tools in mid-project 3%

# Is Software Project Management Improving?

- In industry, are project success rates improving?
  - Failed IT project rate: 14% in 2016 (vs. 23% in 2000)
  - Why?
- Maturing project management skills and tools
- Digital convergence and project prioritization
- Talent and leadership, project managers engaged
- Established PMOs
- Executive sponsors engaged
- Agile methods (71% of organizations)
- <https://www.cio.com/article/3174516/it-project-success-rates-finally-improving.html#:~:text=According%20to%20the%20PMI%20research,14%20percent%2C%20the%20research%20revealed>



# Rescuing a Project

- How bad can it be?
  - No one has any idea when the project will finish, and most people have given up trying to guess.
  - The product is laden with defects.
  - Team members are working excessive numbers of hours— 60 hours per week or more of involuntary or peer-pressure-induced overtime.
  - Management has lost its ability to control progress or even to ascertain the project's status with any accuracy.
  - The customer has lost confidence that the development group will ever deliver the promised software.
  - The team is defensive about its progress.
  - They feel threatened if anyone outside the team suggests that the project might be in trouble.
  - Relations between developers, marketers, managers, quality assurance, and customers are strained.
  - The project is on the verge of being cancelled; customers and managers are actively considering that option.
  - The morale of the development team has hit rock bottom. The fun has gone out of the project, and the team members are grim.
- Steve McConnell from Rapid Development

# Rescuing a Project

- Your options
  - Cut the size of the software so that you can build it within the time and effort planned.
  - Increase the process productivity by focusing on short-term improvements.
  - Face the fact that the software will not be ready on time, slip the schedule, and proceed with damage control, possibly including canceling the project.
  - Combine all these: Drop a few features, increase productivity as much as you can, and slip the schedule as needed.
- Steve McConnell from Rapid Development

# Rescuing a Project

- First Steps
  - Assess the situation
  - Theory-W (or bail)
  - Prepare yourself to fix the project
  - Ask the team what needs to be done
  - Be realistic
- People
  - Restore group morale
  - Clean up personnel or leadership problems
  - Beware adding people
  - Focus people's time, have them pace themselves
  - Everyone responds differently
- Process
  - Identify and fix classic mistakes
  - Fix broken development processes
  - Create detailed miniature milestones and schedule
  - Track meticulously
  - Record reasons for misses
  - Recalibrate every one or two weeks
  - Don't commit until you can
  - Manage risks carefully
- Product
  - Stabilize requirements
  - Trim features
  - Assess politics
  - Take out bad/trouble code
  - Reduce defects, track progress
  - Get to a known good state
- Steve McConnell from Rapid Development

# Things you can do as a developer to improve outcomes

- Own

- Coding Standards
- Test-First Coding
- Design for Change
- Incremental Integration
- Throwaway Prototyping
- Up-front Design

- Contribute to

- Find Error-prone Elements
- Defect Tracking (full lifecycle)
- Daily Build/Smoke Test
- Architectural Design
- UI Prototypes
- Evolutionary Delivery
- Source Control

- From “Low-Hanging Fruit”  
Construx

# Things you can do as a technical lead to improve outcomes

- Own

- Coding Standards
- Test-First Coding
- Design for Change
- Incremental Integration
- Throwaway Prototyping
- Up-front Design
- Find Error-prone Elements
- Defect Tracking (full lifecycle)
- Daily Build/Smoke Test
- Architectural Design
- UI Prototypes
- Evolutionary Delivery

- AND

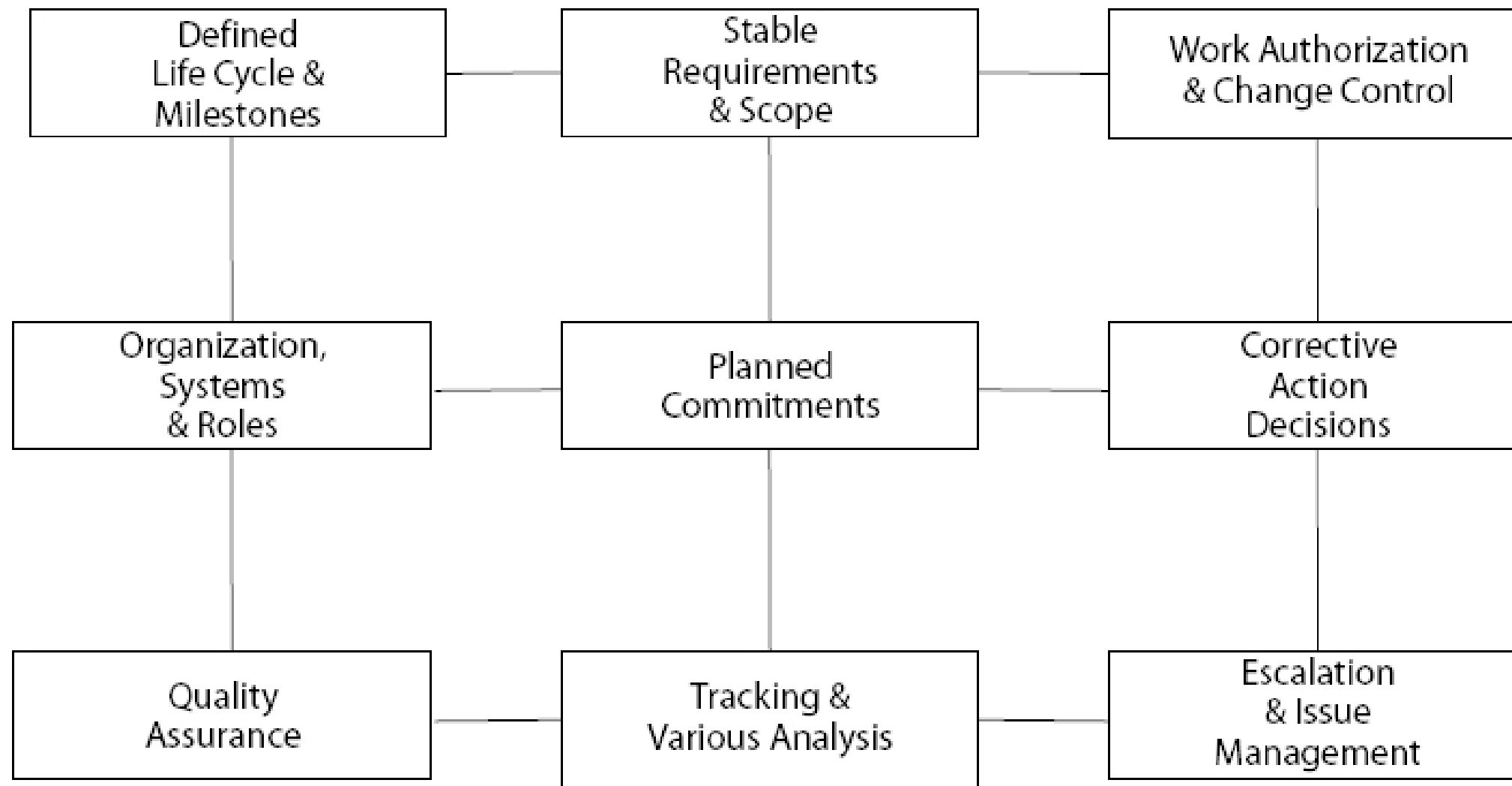
- Lifecycle model
- Inspections
- Requirements Scrubbing
- Incremental/Iterative Planning
- Formal Change Control
- Top-10 Risks

- Contribute to

- Source Control
- Miniature Milestones
- Timeboxed Development
- Up-front Requirements
- Up-front Planning

- From “Low-Hanging Fruit” Construx

# Nine Elements to Project Success



- <https://www.pmi.org/learning/library/best-practices-effective-project-management-8922>

# Minimal Project Management – What you “can’t not” do

- Charter – establish sponsor, governance, validity
- Definition – objective, context, goals, deliverables, scope (in vs. out)
- Staffing – roles & responsibilities
  - RACI chart – Responsible, Accountable, Consulted, Informed
- Work Breakdown Structure (scope) & Timeline (schedule)
- Launch meeting
- Weekly status – task status, risks & issues, changes, new information
- Planning to end the project
  - From Bob Lewis’ “Bare Bones Project Management” 2006
- Also “Project Jeopardy” clarity
  - Thanks to Russ Miles

# Project Status: Constraints and Control

- Time
- Cost
- Scope
- Quality
- The Triple Constraint →

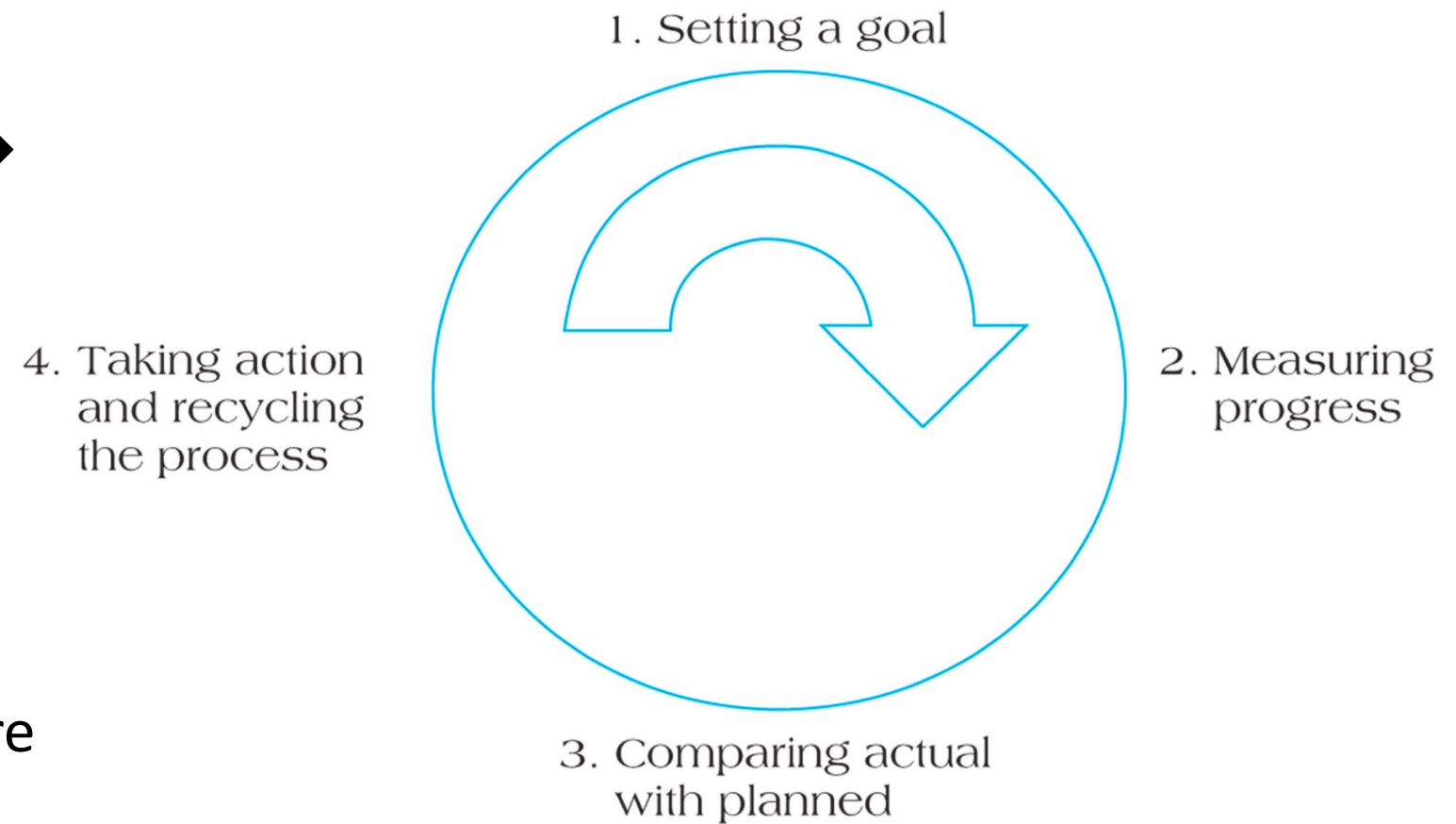


- How do we measure project progress?
- How do we keep stakeholders informed?
- How do we know if a project is going off track?
- Must measure, control, and report:
  - Spend vs. budget (not really an issue for PMP)
  - Delivery dates vs. schedule (project planning and tracking)
  - Quality results vs. requirements (requirements management)



# Project Status: Control Cycle

- Plan/Do/Check/Act Cycle →
- Project Rhythm is a key
- Project manager selects
  - What to measure
  - When to measure
- Compare measure to plan
  - Where are we now - measure
  - Where should we be – plan
- Gaps here require corrective actions and disclosure
  - Keep status open
  - Maintain trust



# Weekly Project Status Form

- Every week
  - Complete the form
  - Review at the status meeting with Bruce or Preethi
  - Review/share with the sponsor
  - Turn in the Project Status Form on Canvas
- Elements
  - Status Summary: Red (off 10%) Yellow (off 5%) Green (on track)
    - Assess Scope, Schedule, Risk, Quality
  - Progress Last Week: What got done last week along with comparison to the plan?
  - To Do This Week: What will be done between now and next status meeting?
  - New Issues, Risks, Changes, Blockers: New issues to raise, any other concerns?
  - Update on Last Weeks Issues, Risks, Changes, Blockers: What did you do to address last week's issues?
  - Other: Anything else to report
- Once WBS/Schedule completed, can begin to report on deliverables
  - Or cards/stories for Scrum rhythm

# Project Status Form

- Suggested template in Class Files on Canvas
  - Project Status Form.doc

Like most project deliverables, when you share this with the sponsor, make sure to ask if this is sufficient or if they'd like to see other status tracking in another form...

## CU CS Professional Master's Project: Project Status Report

Project name: [NAME]

Date: [DATE CREATED]

Prepared by: [WHO WROTE THIS]

### STATUS INDICATORS

SCOPE

SCHEDULE

RISKS

QUALITY

### Status Summary

Scope:

Schedule:

Risks:

Quality:

**Progress Last Week:** What got done last week along with comparison to the plan?

1.

2.

3.

**To Do This Week:** What will be done between now and next status meeting?

1.

2.

3.

**New Issues, Risks, Changes, Blockers:** New issues to raise, any other concerns?

1.

2.

3.

**Update on Last Weeks Issues, Risks, Changes, Blockers:** What did you do to address last week's issues?

1.

2.

3.

**Other:** Anything else to report

# Other Project Concerns

- What are some critical performance indicators that are hard to measure, and therefore hard to report on?
- How are people feeling?
- How are people getting along?
  - In businesses, this can often lead to turnover
- How is the sponsor feeling?
  - Do they seem satisfied? Engaged?
- Am I reporting on stuff that nobody cares about?
  - Focus on what's important and really happening
- Is there anything the team needs Bruce or Preethi need to do?

# Overall PMP Schedule

- Week 3: 9/7

- Initial and final team assignments
- Bruce will notify sponsors of team assignments
- Bruce will send out NDAs for signatures – Due Wed 9/16
- Teams should hold an initial meeting and discuss team roles
- Charter and project briefs assigned/initial development
- Thursday speaker (attendance tracked) – Amy & Rae
- Request an introductory meeting for your team and the sponsor

- Week 4: 9/14

- First meetings with sponsors – if you can share initial charters/project briefs all the better
  - Review any process, deliverables, or tool requirements they may have
- First meetings with Bruce/Preethi for project status updates
  - Begin using status update forms, share with sponsors
- Charter and project brief due, charter submitted for sign off by sponsors
  - Interim by 9/18, Final signed by 9/23

# Overall PMP Schedule

- Week 5: 9/21
  - WBS & Requirements
- Week 6: 9/28
  - WBS & Requirements – pass 1 reviewed by sponsor
- Week 7: 10/5
  - WBS & Requirements
- Week 8: 10/12
  - WBS & Requirements – pass 2 reviewed by sponsor
  - Midterm exam (take home)

# Overall PMP Schedule

- Week 9: 10/19
  - Begin 1<sup>st</sup> 2 week Scrum sprint – Architectural/System Design
- Week 10: 10/26
  - Scrum
- Week 11: 11/2
  - Begin 2<sup>nd</sup> sprint – Design/Prototyping
- Week 12: 11/9
  - Scrum
- Week 13: 11/16
  - Begin 3<sup>rd</sup> sprint – Design/Prototyping
- Week 14: 11/24 (off 11/26-11/27)
  - Close 3<sup>rd</sup> sprint
- Week 15: 11/30
  - Final sponsor and in-class presentations
  - Assessments: Instructor, GSS, sponsors, peer
- Week 16: 12/7
  - Final exam (take home)

# Team Roles

Set these roles in  
your first team  
meeting...

- You can adjust this
  - It may be individuals wear a couple of role hats
  - It may be your team doesn't need the role or needs different ones
- Everyone should be responsible for some code development
- Potential roles:
  - Project or Team lead – point of contact for the sponsor and class staff
  - Technical or Architectural lead – ensure project technical thoroughness
  - Source control lead – merging, code reviews
  - Test and Quality lead – TDD, unit test, integration/system test, test plans
  - Documentation lead
  - Deploy/install lead
  - Communications
  - Requirements
  - Etc.

Team Size	2	3	4	5	6	
Lead	Project	Project	Project	Project	Project	Deliverable Tracking, Sponsor Contact
Roles	Technical	Technical	Technical	Technical	Technical	Architecture
	Source	Source	Source	Source	Source	Source Control and Merging, Code Rev
	Test	Test	Test	Test	Test	Test and Quality
				Build	Build	Versioning, Deploy/Install
					Documentation	Control/Review Project Docs
	(Can switch at Midterm if needed)					



# Communicating with Project Sponsors

- Once the teams are formed, you'll set up a communications plan with your sponsors
  - Your team should plan to **meet with your sponsor this week**
  - The initial meeting is introductory, but you should share:
    - You're developing a Charter and Project Brief for their review
      - If you can show a preliminary, great
      - If they have a charter format they'd like you to use, great
    - You need to know about any requirements they have for tools, processes, or deliverables – and need to review and approve these for you
- Please remember, if you communicate via e-mail, cc both Bruce and Preethi, to allow us to monitor written communications between teams and sponsors
- For our projects, the sponsor owns all Intellectual Property (IP) rights resulting from the master's project
- Do not discuss, reveal, or distribute project materials outside of your team, sponsor, and the class staff without express permission from your sponsor

# Non-Disclosure Agreements (NDAs)

## Appendix C – Student Acknowledgement

### ACKNOWLEDGEMENT OF STUDENT OBLIGATIONS TO COURSE PROJECT SPONSOR Computer Science Professional Masters Project, CSCI5040/5050

- Once teams are assigned, Bruce will send NDAs out for your signature, they will then be provided to CU and the Sponsor
- You must sign this agreement to work on the projects
- Basically, because all our projects will have IP owned by the sponsor, you may not disclose any information about the project to anyone outside of the PMP class without your sponsor's express permission
- Standard stuff, CU protects your interests

I, \_\_\_\_\_ [PRINT NAME], Student, having enrolled in the University of Colorado Boulder's CSCI5040/5050 academic course, acknowledge that I will be permitted to work on a related industry-sponsored Graduate Design Project ("Course Project") only upon accepting certain responsibilities related to the sponsor's information and sponsor's ownership of results and deliverables of the Course Project.

By signing below, Student acknowledges and agrees to the following:

#### ACKNOWLEDGEMENT

Student wishes to participate in Course CSCI 5040/5050 (the "Academic Course") at University under the direction of the University's Academic Course Professor, Bruce Montgomery ("Course Director"), the purpose of which is to provide Student with experience applying skills and knowledge in the field of Computer Science to a technical problem in the field.

\_\_\_\_\_  
SPONSOR NAME (hereinafter "Sponsor") has entered into an agreement with University dated \_\_\_\_\_, titled Graduate Design Student Project Fixed Price Agreement, (the "Project Agreement") for providing financial and mentoring support for students enrolled in the Academic Course to work on a particular technical problem described as follows:

\_\_\_\_\_  
PROJECT NAME (hereinafter "Course Project").

While enrolled in the Course, Student wishes to devote his/her efforts to working on the Course Project. Students working on the Course Project may expect to receive the following educational benefits:

1. Student will have the opportunity to work on a real industry project.
2. Student will have the opportunity to interact and learn from industry experts who are mentors for the Course Project.
3. Student will have the opportunity to work with state of the art tools located at industry sites and within the Department of Computer Science.
4. As a result of interactions with industry sponsors, Student may obtain opportunities for professional advancement or employment.

In return for such educational, experiential, and other benefits of participating in the Course Project, Sponsor shall be entitled to take ownership of any New Intellectual Property. "Intellectual Property" includes: a) any art or process, machine, manufacture, design or composition, or any new and useful improvement thereof, or any variety of plant, which is or may be patentable under the patent laws of the United States ("Inventions"), and b) original works of authorship fixed in a tangible medium of expression under the copyright laws of the United States ("Works"). "New Intellectual Property" shall mean all Intellectual Property conceived or first reduced to practice or made during the performance of this Agreement by the student(s) working on the Course Project.

#### AGREEMENT

##### 1. Proprietary Information

- a. "Proprietary Information" shall mean all information provided by Sponsor to Student and clearly identified by conspicuous legend as proprietary or confidential by the Sponsor at the time of disclosure. In order to be considered proprietary, information disclosed orally or in any other transitory medium must be identified to the Student as proprietary at the time of disclosure and summarized in writing within thirty (30) days after such disclosure. Specifically excepted from this definition of Proprietary Information is all information:
  - (i) known by the receiving party at the time of disclosure;
  - (ii) publicly disclosed except by breach of this Agreement;
  - (iii) rightfully received by the receiving party from a third party without an express obligation of confidence;
  - (iv) independently developed by the employees or agents of either party without any use of Information provided by the other party; or
  - (v) is required by law or regulation to be disclosed.
- b. Except in connection with and in furtherance of Student's participation in the Academic Course and work on the Course Project (including but not limited to communications with Sponsor Mentor(s), University instructors and mentors, faculty advisors and other students enrolled in the Course Project), Student shall not at any time or in any manner use, copy, disclose,

# Regular Project Status Meetings

- With Preethi
  - Inspiring - Site Rework
  - Helping - Web Integration
  - Double Helix - Microscopy Algorithms
  - Inovonics - ADL Algorithms
- Schedule a regular Zoom meeting for your entire team with Preethi or I starting the week of 9/14; start with a 30 minute meeting, we'll get it down to 15.
- With Bruce
  - Status - Conversational Interface
  - Status - Psychological State Algorithm
  - Trimble - IoT Network
  - Edwards - Network Simulation
- Attendance by whole team required
- We'll discuss status update submissions for these meetings soon

# Next Steps

- No class on Thursday – get that Charter done!
  - Bruce is out of town on Friday – delayed responses
  - Next Tuesday 9/22: WBS, Schedules, Tracking, Trello
- Discussion topics up on Piazza!
  - Please try to visit weekly for comments (and participation grades)
- Teams should be working Charter and Brief
  - Use project descriptions to start, capture things you don't know or need to find out from the sponsor
- Initial meetings with sponsor this week
  - Agree on communication for future work, meeting rhythm
  - Review plans for tools, processes, deliverables
- Confidentiality/IP Agreement – signed and completed – due Wed 9/16
  - Will be sent to sponsors and CU project office
- I'd like to see you close on the Charters and Briefs as soon as possible
  - Interim versions by Friday 9/18
  - Charter signed off by Sponsor by Wed 9/23
- Teams meet with me or Preethi 30 min each week (eventually, we'll get that to 15 minutes) (tracking attendance)
  - Starting this week! Bring completed Project Status Forms, turn in weekly!
- Always cc Bruce & Preethi on sponsor e-mails
- Speaker 9/24 – former Buff CS Grad student panel (tracking attendance)
- Preethi and I are available for questions or other support