
CS 383
Software
Engineering
Team
Leads

Contents

I. Introduction 2
1. Software Engineer employee of the company. 2
2. Team lead for your project group. 5
3. A software organization specialist. 7
II. Software Organization Specialists 8
1. IT Manager 8
2. Software Architect 12
3. Quality Assurance Manager 14
4. Project Manager 16
5. AI specialist 18
6. Version Control Manager 19

If your team only has 5 team mates then lose team lead #4. In that case, team lead #1, post the sample Gantt chart, and each person use it directly as is.

I. Introduction

In the coming weeks you are going to be members of an organization. In fact each of you will be filling *several* rolls in the organization:

- Software Engineer employee of the company.
- Team lead for your project group.
- A software organization specialist. (See the team lead titles Part II of this document.)

Each of these rolls will have specific responsibilities and deliverables.

1. Software Engineer employee of the company.

The software Engineer is the generic employee in an organization doing the work of writing actual code. They have gone by many names over the years: coders, software designers, software engineers, software architect, etc. I cannot definitively tell you why the name changes so often, but my personal theory is that each time we assign a name to this role, everyone who uses a computer for even simple things like word processing starts to say that they are also a "software engineer", so the name keeps changing to differentiate what the professionals are doing behind the scenes to make the upfront work of the average computer user easier.

Since I am unsure what this roll will be called when you read this document, I will refer to you as software engineers, making a distinction between computer science (the theory and fundamentals) and software engineering (the practical issues of developing useful software). The focus of this course is software engineering.

Your primary deliverable for this role is your performance on the oral exam. The oral exam mark consists of two components: Contributions and Technical. These two marks will be

Your primary deliverable for this role is your performance on the oral exam. You will be marked primarily on how well you wrote the feature you defined in your Champion document.

multiplied together to give you a final mark. Therefore you need to do well in both areas in order to get a good mark.

The technical portion will consists of some coding techniques that you will learn about during the course. These will vary from semester to semester, so refer to the oral exam marking guide to see what they are this semester. You will want to integrate these into your code as you are writing it.

The contribution portion of oral exam mark will come from developing the feature that you defined in your Champion document, keeping your code integrated with your team's code and from your team member deliverable from each week.

As a contributing software engineer, you are responsible for developing the feature that you defined in your Champion document during the System's Analysis phase of this course. This is where the bulk of your mark will come from. The remainder of the mark (staying integrated and doing the team member portion each week), comes from how well your developed code integrates with your team's code.

In the following section we will discuss the role of the team lead, but as a software engineer, your role is to follow the lead of your team lead and ensuring that you give them the information they need to do their job in a timely fashion. Remember, you will be team lead yourself soon, so it is in your best interest to make the team lead's job as easy as possible for your group.

One of the biggest ways to help your team lead is keeping your code integrated with your team's code and keeping your team's Gantt chart up to date on your progress. Each week your team lead will be giving a status update on your project showing what each person has worked on. You are responsible for ensuring that your latest code is merged in and that your team lead knows what you have worked on in time for your team lead to give the presentation.

For each Software Organization Specialist section below there is a team member's deliverable. Ensure that you finish each of them in time for the team lead to integrate the results into the presentation.

Keep in mind that being a software engineer will probably be your primary role after you graduate.

2. Team lead for your project group.

For about a week or more, you will be the leader of your team. During this time you are responsible for managing your team, ensuring that is everything on schedule.

Your primary deliverable is giving an update on your team's current status at the end of your time as team lead. See the schedule to determine when you week as team lead ends.

Your primary deliverable for this role a 15 minute (approximately) individual presentation where you give an update on your team's current status. In order to give an effective presentation you will need to know what each person in your team has been working on.

Your status report presentation will be marked on both style and content. See the marking guide for the specific areas on style, but for content your presentation will include:

- Introducing yourself, the name of your company and the name of your game.
    - Note: "branding" (making your product's name sound familiar) is extremely important to marketing. Put your company logo on every page of every presentation.
- Demoing the latest (running) version of your team's code.
    - Team lead number 1 has no running code yet, so just show where the code is in the repository and each person's initial contribution.
- Showing your team's Gantt chart. Pointing out for each person what they have completed since the last presentation, what they are working on now and what they have left to complete.
    - Is each person on schedule?
    - What is the cost so far?
    - What is the estimated cost to completion?
    - Are there any "road blocks"?
    - For each person possible offer tangible proof that they completed what the Gantt chart says they completed that

week. For example if the Gantt chart says they completed a new feature for the game, show the feature in a running version of your code.

- Showing the demo of your automated tester with your merged Gantt chart. (Week 3 and later)
    - For each feature completed this week show the automated test case that confirms that it is working as expected.
- Showing how your software development specialist deliverable has been merged with your team's project.

3. A software organization specialist.

In order to ensure consistency in marking and learning outcomes in this course the bulk of your mark comes from each student doing roughly the same things as every other member of the class. However, for this portion of the mark you will be providing a unique service to your team that will not be duplicated.

There are 6 rolls that need to be performed:

- IT Manager
- Software Architect
- QA Manager
- Project Manager
- AI Specialist
- Version Control Manager

Your primary deliverables for this role are an actual deliverable as defined in Part II of this document, and a group talk on that deliverable given the lecture before your team lead talk.

These roles are expanded in more detail in Part II of this document, and they are in the same order that Team Leads are assigned. (I.e. Team Lead 1 is also the IT Manager.)

Each of the other teams has someone preforming the same role as you. You will team up with them to define an industry standard for your role, and together your will give a short presentation the lecture before your team lead talk.

Once your colleagues have defined the industry standard using the tools I have made available for each team lead, you then have the responsibility of integrated it into your project team's system. Part of your team lead talk must include a report on this integration.

Look closely at Part II for a detailed breakdown of what is expected. Also look at what is required for each of the other specialists since the team member deliverables will count towards your contribution marks in the oral exam.

II. Software Organization Specialists

1. IT Manager

Information Technology (IT) managers are responsible for planning, coordinating and directing activities related to computer and information systems of a company. From the hardware to software to the network of an organization, IT managers are in charge of their installation and maintenance.

For this class you are responsible for getting the group to the point where each person can write code on their own hardware and contribute it to the team's project.

Start by taking an inventory of the hardware resources available within your team. Does each person have a laptop capable of running your team's code? If not, consider going to a more light weight software solution or find a way to make the necessary resources available to each person. (For example if someone does not have a laptop powerful enough to run Unity but they do have a PC set up a system where they can write the required code at home, but borrow a team mate's lap top for presentations).

Note that you are the first team lead, so you are the team lead throughout the Systems Analysis phase of this project. Therefore you are responsible for any technical requirements for the Systems Analysis presentation. Make sure in advance that you know how to connect your presentation to the university's system. (Even if you are not team lead 1, you will still need to know this for the individual presentations that come later, but for the team lead number 1 this skill become important sooner.)

Team member deliverables:

Most importantly, by the time your team lead presents you should have a copy of your team's basic project loaded onto your personal computer and you should know how to update your team's code.

You should have the following skills mastered:

- Adding to your code repository
- Updating your team's Gantt chart.
- Communicating electronically with your team

Your team lead has created a doc folder on GIT. By the time you give your System's Analysis Presentation I should find in this folder:

- Your group's RFP
- Your Champion document (prefixed with your initials)

In your doc folder, by the time team lead 1 presents I should also find

- Your feature's class diagram (prefixed with your initials)
    - It must include inheritance and aggregation
- Your feature's sequence diagram (prefixed with your initials)
    - It must include at least 3 classes

Your team lead has created a src folder on GIT. By the time team lead 1 presents in it I should find:

- At least one file code file for your feature that you intend for team lead 2 to integrate into the initial build. (It does not have to be integrated at this point).

Your team lead has also created a presentations folder in the doc folder. Any PowerPoint you make for any presentations will go in that folder.

Software Development Specialists deliverables:

As a group:

- A how-to manual of how to get the initial repository set up, adding code to the repository (GIT ignore will be useful), and handle merge conflicts. Include topics such as:
    - Start-up (Where to download GIT and clone a repository)
    - Workflow (check-out, edit, merge updates (if needed), check-in (only of files that are not ignored))

    - Disaster recovery (How to revert to the last confirmed check-point)
- Put a copy of this file in your doc folder on GIT

As an individual for your team:

- Create a Code Repository (GIT?) with the following folders:

  If using QT and C++:

  doc/

  presentations/

  inc/ - With a sub-folder for each team member

  src/ - With a sub-folder for each team member

  tst/ - With a sub-folder for each team member

  If using Unity and C#:

  doc/

  presentations/

  Assets/

  src/ - With a sub-folder for each team member

  tst/ - With a sub-folder for each team member

  prefabs/ - With a sub-folder for each team member

- Gantt chart repository (Google docs?)

  One of your team mates is responsible for creating the document itself. You just have to give them a place to put it where everyone can access it without overwriting each other.

- Communication (Discord) Set up some communication channel for your group (i.e. Discord, Slack, etc.)

- Many of the above activities will require an invitation to each group member and to the professor in order to begin using them.
- Create a basic Unity project (on GIT) that each team member can add to as needed.
    - Note: No real content is needed. You just need to set up the frame work that will be added to.
- Put all group documents that were included in the SA demos in the doc folder
- Ensure each team member understands the repository, including why NOT to make private branches. Let them know that if they ignore this and make a private branch anyway that only code that is integrated with the main branch will count towards their grade.
- Going forward, be prepared to troubleshoot GIT issues for your team.

Software Development Specialists Presentation:

- Demonstrate to the class doing the things in your how to manual. Create a GIT repository, create a simple Unity project (no content needed)
- Have at least two team members (independently, one after the other) get the repository, make a few changes and upload the results. Confirm that you can see the previous changes.

Add to your Team Lead Presentation:

- Show the directory structure on your GIT.
- Show each team members initial contributions to GIT in both the doc folder and in the src folder

2. Software Architect

A software architect is an expert at software development who makes high-level design choices. For this class the software architect is responsible for creating a minimum viable product for your game. This is a first release that – although it is not sophisticated – is still a playable game.

Team member deliverables:

In the src/name directory put your basic C files with stubs for most functions. (Note: In this directory EVERYTHING goes into your sub-folder. No one will look at this except you and the person your company hires 10 years from to take over your code. Target any documentation to help future developers. I.e. if you are using third party code, think about what happens if the company that made that code no longer exists.)

- Give team lead 2 a list of your .cpp or .cs files and ensure you know how to add/delete to your variable.
- Work with your team lead to determine 1 or 2 functions from your section that are needed for your team leads minimum viable product. Complete those functions.
- Ensure you understand the process of updating, compiling and running your part of your team's code. (You can NOT edit code owned by a team member. Only your own).

Software Development Specialists deliverables:

As an individual for your team:

- A minimum viable product: a program in Unity that is a playable game that integrates code from each member of your team.
    - It should have a main character that is controllable by a user
    - It should have at least 2 NPCs (Non-player characters) that do something in response to the main character.
    - Each person's code must be called at least once in the demo

- Ensure that each team member can download the whole code and that it compiles on their system.
- Make a blank stress test level for team lead 3 to add to later.
- Going forward be prepared to handle issues with the main compile process.

Software Development Specialists Presentation:

- Following any of the many tutorials on-line create a simple 3D game in Unity from scratch.

Add to your Team Lead Presentation:

- Showing the running code is part of the presentation anyway, but make yours more focused. Point out when each person on the team's code is run and what the impact is on the minimum viable product.

3. Quality Assurance Manager

Software Quality Assurance (QA) managers ensure that the product meets a required threshold of acceptability. They build and run a series of tests on the project throughout the development stage of the project and even after the project is released.

The primary goal of Quality Assurance is to find bugs and report them back to the people responsible for the code the bug is in.

In a business, QA personnel would do a mixture of automated tests (so regression testing is possible) and manual tests using the product in ways that the original developer might not have thought of and look for any anomalies. This is one of my favorite jobs since going out of your way to try to break the code is literally in your job description.

For this class we will focus on automated testing, which the QA manager will set up and each team mate will add to.

Team member deliverables:

Note: All tests must be fully automatic and not require user input except to start the test case. Each test case must return a clear pass fail (or a failure point for stress tests).

- Define at least one stress test case and add it to the group's stress tests. Each stress test must define a failure condition. The stress test should start at a success position, then gradually increase the stress until a failure is detected. It should then return the value where the failure occurred.
- Create at least 2 initial boundary tests for the automated testing system that your team lead will set up. Be purposeful in your testing. Think about your boundary conditions.

Team lead 3 will be responsible for tying it all together, but each of you will have to come up with your own tests.

Software Development Specialists deliverables:

As a group:

- Create a how to document for using Unity Test Runner. It should include:
    - How to create a new boundary test case
    - How to create a new stress test
    - How to run the tests and interpret the results.

As an individual for your team:

- Referring to the notes on BBLearn, use Unity Test Runner to create a test plan for your group.

Software Development Specialists Presentation:

- Demonstrate all the aspects in your how to manual.

Add to your Team Lead Presentation:

- Showing the test plan is part of the presentation anyway, but make yours more focused. Show each of the test cases one by one showing what they are testing for. In future presentations your class mates will just have to show the tests and the pass fail. You should show:
    - The source code that runs each test
    - What it looks like when running
    - The rationale of what person was trying to achieve.

4. Project Manager

A project manager is responsible for organizing and motivating the project team. The project manager monitors the progress of the project, analyzes any risks and keeps an eye on the cost both in time and money.

For this class the project managers is primarily focused on creating the Gantt chart using the sample that has been given to you.

The creation of coding standards is usually done by the system's analyst, but in order to balance out the work load it has been moved here.

Team member deliverables:

Add to the Gantt chat of all the work items you have completed so far and what you have left to do (with time estimates in hours).

Be specific about what tasks you will be doing. (i.e. "Searching the internet for graphics", or "Coding layout for level 2", etc.)

You must keep this chart up to date each week.

Software Development Specialists deliverables:

As a group:

- Review the slides on coding standards and at least one other source such as https://google.github.io/styleguide/cppguide.html.
  "The point of having style guidelines is to have a common vocabulary of coding so people can concentrate on what you are saying, rather than on how you are saying it."
- Come up with a coding standard with your group. Include
  - Naming conventions (camelCase, how to choose good names, etc.)
  - Commenting styles (Inline, function heading and file heading)
  - Error handling
  - Indentation
  - Etc.

- Put together a 2-3 page document (mostly examples) and put it in the doc folder on each teams GIT.

As an individual for your team:

- Create and maintain the Gantt chart for your group.
- Create a developer's manual including:
    - Environment setup for your platform (be specific about which version of unity, unreal or QT you used, including a link and download instructions).
    - Using the context diagram, and existing class diagrams, give a high level view of the existing code.

Software Development Specialists Presentation:

- Go over your developer's manual.
- Each person look at the Gantt chart for another team. Find at least three things that look suspicious (i.e. a team member falling behind, a team member whose projected far more/less than the average amount of work, an estimate of time that looks unrealistic, etc.) and give the list to the team lead of the project you are analyzing. On the day of the presentation pretend that they do not already know what you are going to ask them about and ask them to justify the anomalies. (Prepare your answers in advance for the questions you will be asked, by finding out from the person involved why there looks like there is a discrepancy.)

Add to your Team Lead Presentation:

- Go over your coding standards document. Review the changes produced this week only on your team, (or 2 weeks if a participant did not make any changes this week) for each member of your team; evaluate if he or she is following the coding standards you defined.

5. AI specialist

Honestly I was not sure what title to give this role. The primary objective is to make a demo mode of your game so realistically it could even be called Marketing, but then I would not have much to give you for your group presentation.

Team member deliverables:

Continue to develop your feature.

Software Development Specialists deliverables:

As a group:

- Do some research on Artificial Intelligence in video games.

As an individual for your team:

- Create a logo for your team
- Add a demo mode to your game. The demo mode of the game should run without any user input required. It should begin running automatically when no actual game is being played, but stop instantly when there is any user input allowing the user to begin a game. It must include:
    - Works on any level of your game.
    - Main player is playing normally, but no input from the user is needed.
    - Have at least 2 for each level. 1 success path. One (or more) failure paths.
- An 8.5/11 poster describing your game suitable to be used as an advertisement for the final presentation.

Software Development Specialists Presentation:

- Give a presentation on AI in video games. (This is a wide open requirement. Make of it what you will, but make me proud).

Add to your Team Lead Presentation:

- Show your team's poster (soft copy)
- Show your game running in demo mode. (Both the success case and the failure cases for each level).

6. Version Control Manager

Version control is literally what GIT is designed to do. Version control systems like GIT manage changes to source code over time keeping track of every modification to the code.

The job of a version control manager goes beyond the basic functions of GIT. A version control manager is responsible for managing changes to documents, computer programs and all other aspects of a products different releases (or “versions”).

For this class the version control manager is responsible for the final release of your team’s project onto a number of different platforms.

Each platform is a different release, so while I have been insisting that everyone keep all the code in the master branch, you get to break the trend and create a separate branch for each release that you are targeting.

Team member deliverables:

Continue to develop your feature.

Software Development Specialists deliverables:

As a group:

- Choose one of the team’s games that would run well on the Vive or any virtual reality platform you have access to and can bring to class. Update the controls to run with the controls of the virtual reality environment.
- Choose one of the team’s games to run on a cell phone. Update the controls to be haptic controls. (I.e. you would move the main player by tilting your phone the direction you want to go rather than pushing a button.
- Choose a third platform of your choice to deploy a game to.

As an individual for your team:

- A ½ page user’s manual suitable to have on the desks at the final presentations showing a brief description of the game (similar to the description in the poster), and the basic

commands. Do this even if the information is available in the game itself.

- A readme file for the teams at the other sites with instructions on how to load and run the executable of your game on their computer with minimal set-up.
- Create an executable for your contact at the other site to run during the final demo.

Software Development Specialists Presentation:

- Using the tools available, demonstrate how to deploy code to different platforms, and discuss the issues involved with each.
- Demo code running on at least three different platforms.
- Choose a volunteer who is not on the team that wrote the code and show them the team's user's manual and have them use it to play a game.

Add to your Team Lead Presentation:

- Show your team's user's manual and readme file (soft copy)