

SPRINT ONE

CS2450-002, Team 2

Cody Strange-*Scribe and Information Manager*

Ethan Taylor-*GUI Developer*

Jaden Albrecht-*Team Manager*

Tyler Deschamps-*Chart and Milestone document builder*

Jordan Van Patten-*V&V and Tester*

Craig Sharp-*Stakeholder*

Table of contents

Introduction.....	3
<i>Procedures</i>	3
<i>Requirements Gathering</i>	4
<i>Prototype Candidate</i>	5
<i>Resources</i>	6
Agile Methods.....	8
<i>Scrum</i>	8
<i>Crystal Clear</i>	8
<i>Kanban</i>	9
Programming.....	9
<i>Coding Standards</i>	9
<i>Brainstorming</i>	9
Charts/Templates.....	10
<i>Work breakdown structure</i>	10
<i>PERTT Chart</i>	10
<i>Gantt Chart</i>	11
<i>MoSCoW Chart</i>	12
Meeting Logs.....	13
<i>Meeting Log#1</i>	13
<i>Meeting Log#2</i>	14
<i>Meeting Log#3</i>	15

Introduction

Procedures

Document versioning

Summary: We currently have an archive folder in our google drive that we put any old documents into when we get a new one. Also in the title's of documents we put _v(x) for whatever version of that document is passed the initial version of the document.

Scheduling

Summary: We have the given deadlines from the customer for scheduling when our sprints need to be completed by. Jaden Albrecht is our team manager and schedules our meetings with the stakeholder. We have two meetings scheduled each week at 7pm on Monday and Friday.

Verification & Validation

Summary: We have processes for both verification and validation. Both are in early stages of implementation and are likely to change as we see what works and what doesn't as the sprints progress.

- Verification: We meet the Friday after the sprint begins and discuss our initial thoughts on the requirements of this sprint. We then have the weekend to mull over our ideas and figure out what we missed. Then we meet back on Monday and confirm what we agree needs to be done. We then schedule a meeting with the stakeholder so that we can get a final confirmation on what all needs to be done.
- Validation: We have a V&V folder that when we finish a document we submit it into that folder. Jordan then validates that the documents meet the proper requirements and are ready to be reviewed by the professor. Jordan then either sends them back to us to revise or to the review folder for the professor.

Timekeeping

Summary: We have the Gantt chart and the Burndown chart that shows the estimated time and the total time that is spent on the sprint, and tasks in the sprint. We also have a text message group that each day we send all the tasks we worked on and the amount of time we have spent on them.

Requirements Gathering

Identify Customer

Summary: Currently it is unknown exactly who our customer is. While we know it is Craig Sharp, we do not know enough about him to identify him. We will have to have a stakeholder meeting in order to get the required information.

Requirements

Summary: Requirements gathered from meeting with customer, creating MoSCoW charts, research. Requirements have been split between functional and non-functional. There are also the must-have requirements and the change orders that have been requested.

- Must haves:
 - Database Merging: recognize when there are incomplete fields and making a flag pop up that says “this is an incomplete employee”
 - Add employee: Page(B) of the GUI, Admin access, button on page(A), input all required fields and add employee to database
 - Edit employee: Page(C) of the GUI, General access, button on page(A), list of employee information to edit(First name, Last name, Address, Office phone, Personal phone, Bank info, Office email, Personal email)
 - Edit employee: Page(C) of the GUI, Admin access, button on page(A), list of employee information to edit(SS#, D.O.B, Pay type, Title Dept., Permission level)
 - Search employee by last name or ID: Page(A) of the GUI, General access, button on page(A), if user inputs numbers will search based on ID, if user inputs letters will search based off of last name, pulls up list of names and numbers of every that matches the information input
 - View employee: Page(A) of GUI, Admin access, list of employees names and IDs, when user clicks on name/ID pulls up all information related to that employee, option to view/hide deactivated employees
 - Deactivate employee: Page(D) of GUI, Admin access, button on page(A), input ID of employee that the user wants to deactivate, confirmation message pops up along with employee information to make sure the user wants to deactivate this specific employee
 - Win 10: The program will be able to run on Windows 10
 - Reports: Page(E) of GUI, Admin access, button on page(A), options to produce various reports(pay, reimbursables, employees)
 - Export Reports: Page(E) of GUI, Admin access, button on page(A), option to export any report as a csv
 - Secure records to only admin permissions: Certain information/records will require the user to be flagged as an Admin to see.
 - Intuitive GUI: Mouse over input boxes to see what information is required, help buttons on each page
 - User Manual: Page(F) of GUI, General access, help button on page(A), information on what each page can do
 - Readme.txt: A short description of what data the code contains
 - Simple just download installation that runs: Download software and then the user is good to go

- Garbage proof entries: Checks each field of data to make sure that all information coming in isn't junk
- Warn user of empty data fields: When user adds/edits an employee, issue a warning if any data fields are left empty or incomplete
- Employee can view all personal fields: Page(A) of GUI, when employee logs in with his/her ID that ID's corresponding information is pulled up
- Bug free: Software will go through extensive testing to minimize/eliminate as many bugs as possible
- Requirements for employee information: First name, Last name, Address(use separated fields), Office phone, Personal phone, Emp ID(Specific length, only numbers), Pay type(commission, hourly, salary), D.O.B, SS#(Specific length, only numbers), Start date, Bank Info(if Direct Deposit), Permission level, Title Dept., Office email, Personal email
- Functional
 - Add employee
 - Edit employee
 - Search employee
 - View employee
 - Deactivate employee
 - Reports
 - Export Reports
 - Secure records to only admin permissions
 - Warn user of empty data fields
 - Employee can view all personal fields
 - Requirements for employee information
- Non-functional
 - Win 10
 - Bug free
 - Garbage proof entries
 - Simple just download installation that runs
 - Readme.txt
 - User Manual
 - Intuitive GUI
 - Database merging
- Change orders
 - N/A

Prototype candidate

Jaden Albrecht

Summary: The program can read in a csv file of employee information, each employee is required to have an Id, first name, last name, Address, City, State, zip code, and pay

1/24/2022

classification. The program holds every employee's information in a list that can be accessed to look up an individual's information. The program also requires timecard, receipt, and pay log files in order to calculate each employee's pay. This is the prototype that we chose to use.

Current Features

- Calculate an employee's pay by their total salary, commission, or hourly wage
- Add employees to the database
- Find employees based on their ID number
- Reads in an csv file and out puts a list of employee objects
- Return specific employee information such as ID, name, address(city, state, zip code), classification(hourly pay, commissioned pay, salary pay)
- Alter an employee's classification

Needed Features

- Recognize when there are incomplete fields and make a flag pop up that says "this is an incomplete employee"
- Add an employee's:
 - First name
 - Last name
 - Office phone
 - Personal phone
 - Bank info
 - Office email
 - Personal email
 - SS#
 - D.O.B
 - Title Dept.
 - Permission level
- Search employees by last name
- Deactivate employees
- Export reports as a csv
- Secure records to only admin permissions
- GUI

Resources

Communication

Summary: We primarily use text messaging group to stay in contact with each other outside of planned meetings. We use it to ask for quick updates on tasks that may affect what we are currently doing. MS teams is what we use for our team meetings, we are currently planning to meet twice a week Monday and Friday at 7:00pm. There is also the project email and google drive that is being used to send documents to each other and to store files for the project.

1/24/2022

- MS teams
- Text group chat
- Project email and Google drive

Software

Summary: We are using Lucid Charts as the software to create the PERTT, Gantt, MoSCow, and Work Breakdown charts. Microsoft word is what we are using for documentation for our meeting logs, notes, and sprint documents. We are using Trello to help us organize a to-do list that is similar to Kanban. Python is the programming language that we will be working in, and we are still not sure whether we will be using Thonny or Visual Studio Code as the IDE. Tkinter will be used for the GUI. We will be using a group Gmail to send information to the group and Google Drive to save all of our documents so that the entire group and the shareholder will have access to them.

- Lucid Charts
- Microsoft word
- Trello
- Python
- Thonny/VS code
- Tkinter
- Debugging Software*
- Timekeeping software*
- Gmail
- Google Drive

Research

Sources

- Beginning Software Engineering(chapter 4)
- CS2450 Lecture #6(2/2/2022)
- Customer Meeting #2(2/1/2022)
- CS2450 Lecture #5(1/31/2022)

Agile Methodologies

Scrum

Overview

Summary: As the project sprints are designed around the Scrum methodology, we will be using Scrum in our project management. We will be completing our project in incremental iterations so that at the end of each sprint we will have a fully tested and approved piece of software. We still need to decide exactly what parts of Scrum will and won't work for our team.

- Scrum focuses on breaking large problems into smaller tasks and completing tasks with speed and efficiency.
- Scrum focuses on creating a small workable product soon and building new and more complex versions of that product every sprint.
- Scrum is flexible, allowing constant changes throughout development to meet customers everchanging requirements.

Crystal Clear

Overview

Summary: Along with Scrum will be incorporating part of the Crystal Clear development methodology. Crystal Clear like scrum focuses on frequent releases that get additional features as time goes on. Crystal Clear is very heavy on team communication, it recognizes that in small teams it is valuable to have everyone pitch in on ideas and that rigid structures are not necessary.

- Much more informal methodology
- Expected to deliver production every 2-3 weeks, possibly shorter with non-released development iterations
- Take into consideration what the program will be used for when deciding how strict certain requirements such as testing need to be
- Emphasis on communication and in person collaboration whenever possible

1/24/2022

Kanban

Overview

Summary: The last methodology that we will be using is Kanban, and we will be using Trello to implement this. We use the Kanban board to visualize what needs to be done, what is currently being worked on, and what has been completed.

- Kanban enables us to visualize the work that we are doing today in the context of other tasks
- Kanban board

Research

Sources

Summary: Information was gathered from the following sources

- Beginning Software Engineering(chapter 14)

Programming

Coding Standards

PEP-8

Summary: We will be following the PEP style guide for python code.

Brainstorming

Team Meetings

Summary: We have our team meetings on Friday and Monday at 7pm. This allows us to come together on Friday to begin brainstorming and then we have the weekend to think over the ideas that were presented and meet back on Monday to finish the brainstorming.

Research

Sources

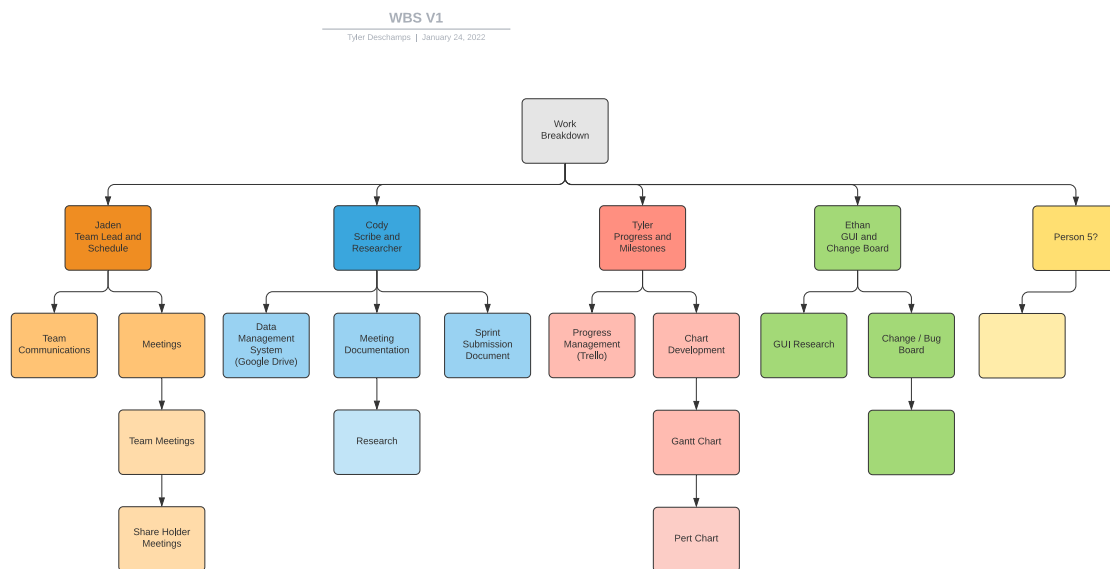
- Python.org

Charts/Templates

Work Breakdown Structure

Chart

Summary: This is a Work Breakdown Structure for our four (soon to be five) members. It details each of our Team members and their roles along with the general tasks that their role comes with.

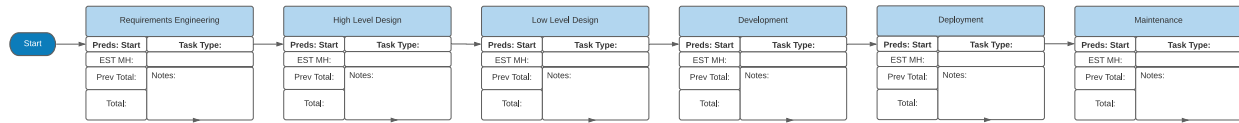


PERTT Chart

Template

Summary: This is a PERTT Chart Template, for each of the six stages. We will be able to fill it out with the estimated time we expect each task to take and come up with a total amount of time for the project

1/24/2022



Gantt Chart

Template

Summary: This is a Gantt chart template; it will list each of the tasks that we will be doing in each sprint.

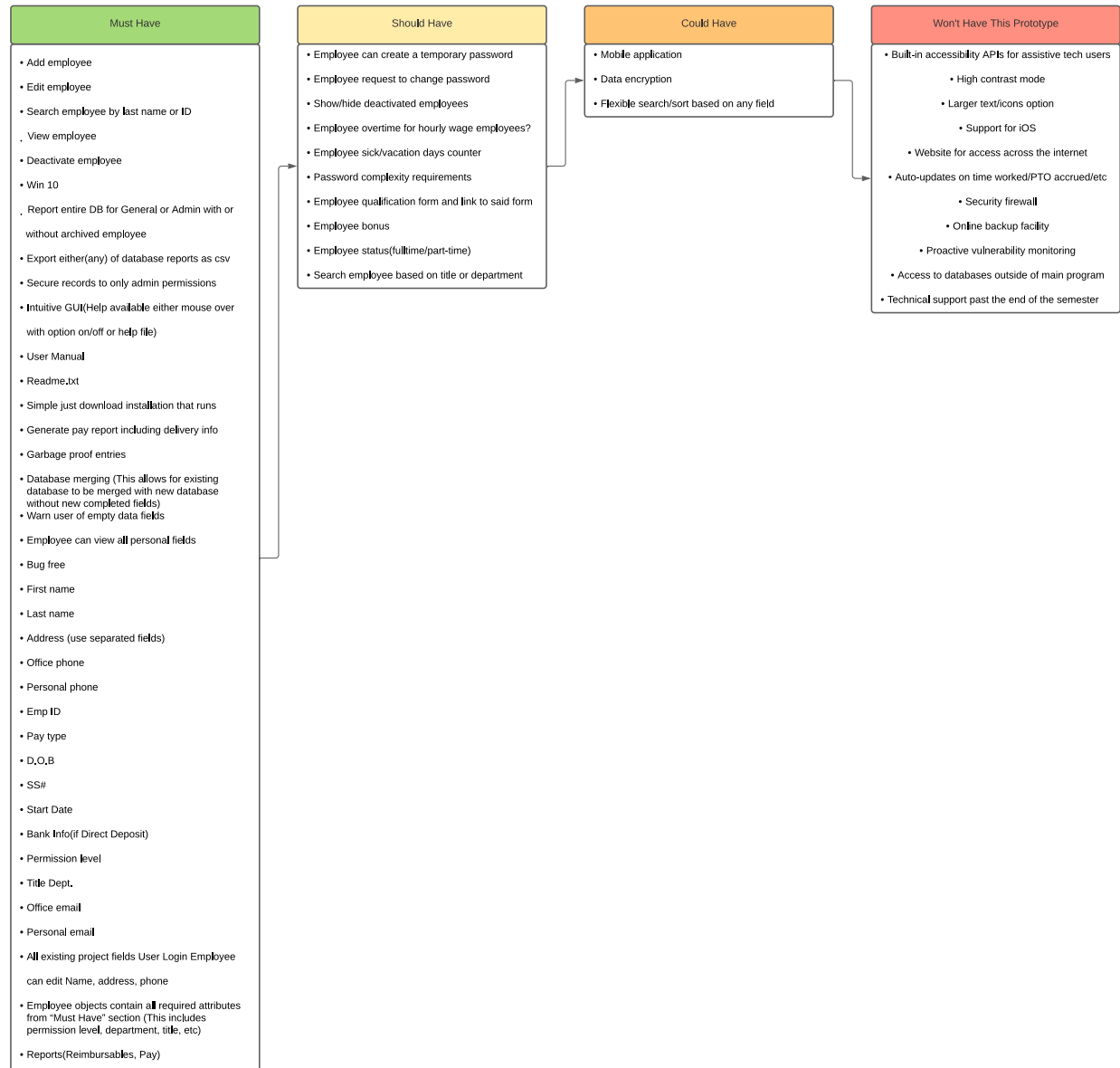
Gantt Chart	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6
Task 1						
Task 2						
Task 3						
Task 4						
Task 5						
Task 6						
Task 7						
Task 8						
Task 9						
Task 10						
Task 11						
Task 12						
Task 13						
Task 14						

1/24/2022

MoSCoW Chart

Overview

Summary: This is a MoSCoW chart template; it will have each team member listing what they believe the product Must have, Should have, Could have, and Won't have. We will each make a separate one and eventually combine/narrow them down to one.



Burndown Chart

Overview

Summary: The burndown chart shows the amount of time expected to be spent on tasks in the sprint against the actual amount of time spent. There are two versions of the chart, the regular burndown chart and the backwards burndown chart.



Research

Sources

- Beginning Software Engineering(chapter 3)
- CS2450 Lecture #6(2/2/2022)

Meeting Logs

Meeting Log#1

Meeting Details

- Organizer: CS2450-002 Team Two
- Date: January 20, 2022
- Time: 7:55pm – 9:07pm
- Location: Microsoft Teams
- Attendees: Ethan Taylor, Jaden Albrecht, Cody Strange, Tyler Deschamps

Topics Discussed

- Set roles:
 - Ethan Taylor: GUI Developer
 - Jaden Albrecht: Team Manager
 - Cody Strange: Scribe and Milestone document builder
 - Tyler Deschamps: Chart and Milestone document builder
- Tasks:
 - Create project email and google drive[Cody Strange]
 - Research Tkinter[Ethan Taylor]
 - Begin MoSCoW charts[Tyler Deschamps]
 - Research templates for PERT and Gantt charts[Tyler Deschamps]
 - Read chapters 11, 12, 13[Cody Strange, Ethan Taylor, Jaden Albrecht]
 - Set up meeting with Professor[Jaden Albrecht]

Summary

- Notes:
 - Everyone is equally confused about the project, but we have found areas that we can start in and are preparing for our first meeting with the Professor.
 - Took stock of talent and skills of teammates
 - Three have python Experience up to CS2420
 - One has had some workings with GUIs
 - Only four out of six members four out team
 - One has knowledge of Trello to use as a project management tool to start us off with
 - Currently everyone plans on participating in the role of being a programmer, this may vary depending on the workload of certain individuals
- Topics for next time:
 - More detailed description of everyone's strengths
 - Has anyone used Scrum?
 - Years of programming experience
 - What does everyone's schedule look like, how much time can everyone expect to dedicate
 - Clarification on roles
- Next Meeting: January 21, 2020 @ 6:00pm, Microsoft Team

1/24/2022

Meeting Log#2

Meeting Details

- Organizer: CS2450-002 Team Two
- Date: January 21, 2022
- Time: 6:00pm – 6:41pm
- Location: Microsoft Teams
- Attendees: Ethan Taylor, Jaden Albrecht, Cody Strange, Tyler Deschamps

Topics Discussed

- Agile methodologies
 - Crystal clear
 - Scrum
 - XP programming
 - Lean
 - AUP
- Requirements for Sprint one
 - MoSCoW charts
 - Gantt charts
 - PERTT charts
- Programming
 - Everyone is a programmer
 - Everyone has a partner who tests their code for them

Summary

- The Agile methodologies we chose to go with were Scrum and Crystal Clear
- We lack the manpower for XP programming
- File storage and organization complete
 - Google drive and project email created
- Ask about slide highlights from professor
- Team specific tools are established
 - Linked everyone to MS teams, counting shareholder
- Weekly meetings twice a week
 - Monday
 - Friday

Tasks

- Install Tkinter-everyone
- Schedule meeting with professor-Jaden Albrecht
- Next Meeting: January 24, 2020 @ 7:00pm, Microsoft Team

Meeting Log#3

Meeting Details

- Organizer: CS2450-002 Team Two
- Date: January 24, 2022
- Time: 7:00pm – 7:55pm
- Location: Microsoft Teams
- Attendees: Ethan Taylor, Jaden Albrecht, Cody Strange, Tyler Deschamps

Topics Discussed

- Final revisions and overview of all Sprint#1 documents

Summary

- **Overview of WBS (Work Breakdown Structure)**
- **Overview of Pert chart**
- **Overview of Gantt chart**
- **Initialized MoSCoW chart**
- **Acquired prototypes**
- **Scheduled next team meeting for January 28th**
- Established role for new team member still undecided
- Began document MoSCoW chart ideas for sprint#2

Tasks

- Submit sprint#1
- Read chapters 5-6
- Begin preparations for sprint#2