**5.** **Project Planning**

Now start the project planning for your system. One way to start this planning is by placing the Inception, Elaboration, Construction and Transition stages on a Project Schedule and then adding activities from each stage. Also start a Risk Plan, consisting of an enumeration of each risk, its probability, impact, transition indicators and risk management strategies. Add risk planning and management activities to the Project Schedule. Next, create a Release Plan and an Iteration Plan for the 1st release, and then add each release and the 1st iteration to the Project Schedule.

As we progress through the lectures and you learn about and develop the other work-products of your project, you will need to update your project plan. For example, once you have developed the Use Cases, estimate the project effort independently using a bottom-up approach as well as the Use Case Points approach, and update your project schedule accordingly.

Similarly, as you understand the requirements and define the architecture you will find that the Risk Plan needs to be updated.

For this section, recommended work-products are:

* Release Plan
* Iteration Plan for the 1st release
* Risk Plan
* Project schedule (note that for this you need the effort estimates)

Update Section 2 as needed.

Evaluation:

* Are Inception, Elaboration, Construction and Transition activities all captured in the Project Schedule?
* Are the elements of the Release Plan the Use Cases and/or Scenarios and or NFR. Is it consistent with the Prioritized Requirements?
* Are the tasks in the Iteration Plan linked to the elements of the Release Plan?
* Is the Iteration Plan complete? Are Exploration, Commitment and Steering tasks all captured in the Iteration Plan?

Notes:

* Items in the Release Plan should be Use Cases, NFR and/or Scenarios/Stories. Note that releases are those versions of the system that go to customers. Hence these should be complete in terms of capability.
* Items in the Iteration Plan are fine-grained tasks derived from Use Cases and/or Scenarios/Stories, and other tasks, such as system set up, any technical spikes, etc.
* Need to have done a UCP estimate + a linear estimate and used that in finding dates and durations for the Project Schedule. Also, compare the results from the 2 types of estimations.
* Project Schedule should have actual dates and durations
* Project Schedule should be represented using a Gantt chart
* Exploration, Commitment, and Steering are phases in each release and iteration. The steps that belong to these phases are of many types and are interwoven into the release or iteration. E, C, and S may also be thought of as the planning, the deciding and doing+monitoring+managing parts of the release or iteration.
* Note that the Risk Plan contains PROJECT risks, not system "risks" - which are really requirements

**5. Project Planning**

**5.1 Release Plan**

* Release 1 (release date September 14th, 2020)
  + Update and optimize the current Emxsys software
    - Make the look and feel more user-friendly, etc.
  + Create a basic customer support portal
  + Set up a notification system for desktop version
  + Reach agreements with local authorities to secure usage and deals
  + Start advertising to increase user base
* Release 2 (release date January 11th, 2021)
  + Add tutorials to customer support portal
  + Make changes based on marketing information, user usage, and feedback
  + Distribute a mobile version of the application
* Release 3 (release date March 1st, 2021)
  + Improve mobile application based on feedback, marketing data
  + Release optimized notification and warning system
    - Location based warnings based on user
    - Impending weather/fire conditions
    - Local Authority updates and information
    - Fire Damage predictions

**5.2 Project Estimation**

**5.2.1 Linear Estimate**

|  |  |
| --- | --- |
| **Tasks and their estimated efforts (hours)** | |
| **Task** | **Effort(hours)** |
| User Interface Update | 150 |
| Customer Support Portal | 140 |
| Desktop Notification System | 100 |
| Reach deals with local authorities | 120 |
| Ramp up advertising | 70 |
| Test functionality of Release 1 | 50 |
| Add tutorials to customer support portal | 200 |
| Improve functionality per feedback, marketing data | 40 |
| Create and distribute mobile version | 500 |
| Test functionality of Release 2 | 80 |
| Improve mobile version per feedback, marketing data | 60 |
| Optimize notification system | 190 |
| Location specific real-time notifications | 70 |
| Weather/fire conditions | 60 |
| Local authority updates and information | 50 |
| Fire damage predictor | 80 |
| Testing functionality of Release 3 | 50 |
| **Total** | 2020 |

**5.2.2 Use Case Points Estimate**

|  |  |  |
| --- | --- | --- |
| **Actors** | | |
| **Actors** | **Actor Type** | **Weighing Factor** |
| User | Complex | 3 |
| Local Authority | Complex | 3 |
| Customer Support | Complex | 3 |
| Market Researcher | Complex | 3 |
|  | **Unadjusted Actor Weights (UAW)** | 12 |

|  |  |  |
| --- | --- | --- |
| **Use Cases** | | |
| **Use Case** | **Use Case Type** | **Weighing Factor** |
| User Interface Update | Complex | 3 |
| Customer Support Portal | Complex | 3 |
| Desktop Notification System | Average | 2 |
| Reach deals with local authorities | Complex | 3 |
| Ramp up advertising | Complex | 3 |
| Test functionality of Release 1 | Average | 2 |
| Add tutorials to customer support portal | Complex | 3 |
| Improve functionality per feedback, marketing data | Average | 2 |
| Create and distribute mobile version | Complex | 3 |
| Test functionality of Release 2 | Average | 2 |
| Improve mobile version per feedback, marketing data | Average | 2 |
| Optimize notification system | Complex | 3 |
| Location specific real-time notifications | Simple | 1 |
| Weather/fire conditions | Simple | 1 |
| Local authority updates and information | Simple | 1 |
| Fire damage predictor | Average | 2 |
| Testing functionality of Release 3 | Average | 2 |
|  | **Unadjusted Use Case Weights (UUCW)** | 38 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technical Factors** | | | | |
| **Factor** | **Description** | **Value(0-5)** | **Weight** | **Product** |
| T1 | Distributed system | 2 | 2 | 4 |
| T2 | Response time/performance objectives | 2 | 1 | 2 |
| T3 | End-user efficiency | 3 | 2 | 6 |
| T4 | Internal processing complexity | 1 | 1 | 1 |
| T5 | Code reusability | 3 | 2 | 6 |
| T6 | Easy to install | 3 | 1 | 3 |
| T7 | Easy to use | 4 | 2 | 8 |
| T8 | Portability to other platforms | 2 | 1 | 2 |
| T9 | System maintenance | 4 | 1 | 4 |
| T10 | Concurrent/parallel processing | 0 | 0.5 | 0 |
| T11 | Security features | 1 | 1 | 1 |
| T12 | Access for third parties | 2 | 2 | 4 |
| T13 | End user training | 5 | 2 | 10 |
|  | **Technical Complexity Factor (TCF)** | **1.11** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Environmental Factors** | | | | |
| **Factor** | **Description** | **Value(0-5)** | **Weight** | **Product** |
| E1 | Familiarity with Dev process used | 0 | 0.5 | 0 |
| E2 | Application experience | 3 | 1 | 3 |
| E3 | Object-oriented experience of team | 2 | 1 | 2 |
| E4 | Internal processing complexity | 1 | 1 | 1 |
| E5 | Motivation of the team | 4 | 2 | 8 |
| E6 | Stability of requirements | 3 | 2 | 6 |
| E7 | Part-time staff | 3 | 1 | 3 |
| E8 | Difficult programming language | 0 | 0.5 | 0 |
|  | **Environmental Factor (EF)** | **0.71** |  |  |
|  | **Adjusted Use Case Points (UPC)** | **39.405** |  |  |
|  | **Total Estimate (UPC \* 30 hrs/pt)** | **1182.15** |  |  |

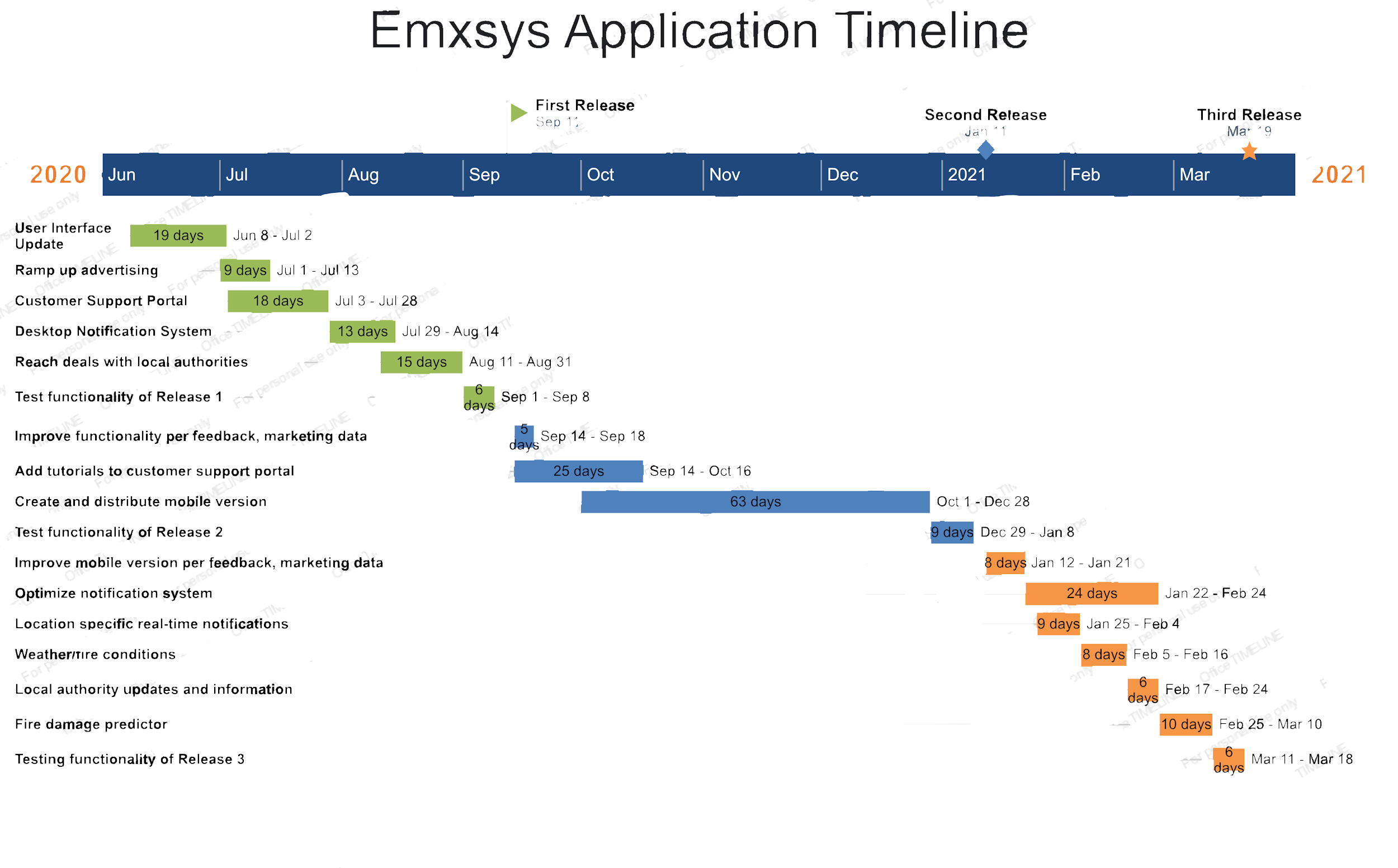
**5.2.3 Comparison**

For the linear estimate, the time requirements were given fairly generously because of the nature and scope of the problem. We felt that it would be best for the estimates to be generous considering the open source nature of the platform and the large amount of changes to be had for the project. With the tremendous task in front of us, we felt it appropriate to leave a large amount of time for any problems along the way. The total estimate for this method was 2,020 hours.

For the use case point estimate, the general guidelines for the use case points estimation technique were used. For these estimates we used generous amounts again to give a fair amount of padding. For this estimate the total was 1,182 hours.

The difference in hours between these two methods is very large. Given the huge disparity, we felt it safer to go with the linear estimate. Although the use case estimate was much more detailed, the extra time for this given project feels like a better fit considering the size of the task.

**5.3 Project Schedule**



**5.4 First Release**

* Functional Requirements
  + System must provide users with up to date wildfire information
  + System must deliver complicated information in a user friendly format
  + System must track user information to better serve customers
  + System should provide an easy avenue to support and help for difficulties
  + System should successfully notify users accurately and in a timely manner in case of emergency
* Non-Functional Requirements
  + System is available 24/7 for reporting.
  + Keep user information private
  + System should not allow malicious users access to location services
  + System should log all unique requests made within 30 seconds.
  + Easy to use interface since users most likely will not be tech savvy

**5.5 Risk Management Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Impact** | **Transition Indicator** | **Mitigation** | **Containment** |
| Low adoption from local authorities/ a lack of deals or agreements | Less income from lack of adoption  Less budget for development | Large amount of contacted government agencies do not agree to use the product | Increase awareness of the usefulness of the project to government authorities | Reduce operating costs to cover for lack of adoption  Hold a meeting to design a new strategy to increase adoption |
| Lack of Open Source Developers | Less expertise from a wide range of volunteer developers  Have to invest more into salaried developers  Project loses valuable open source style of development | Less open source developers begin to submit changes to the project | Improve benefits for open source developers, make it a project worthy of continuing to volunteer for  Break the project into parts so that goals are more easily attainable by the scattered team | Hire more developers so there is a committed staff  Spread the word of the project more so that more volunteers may join |