Project Instructions

It's an individual Project with a Due date of 5 June 2025.
You can find the separated deliverables at Omnivox, and in the outline, late submissions will be accepted! Half of the stacks will be covered during the course the rest you will have to implement through self-study!
Each of the solutions has to be containerized through Docker and hosted in a Cloud of your preference, ideally 2 different clouds so you can compare this aspect too.
I am expecting 24 different business scenarios submitted at Omnivox.
The final submission package should consist of a *repository link* to your *container* with the source of your prototypes, (while during the phase,s you will only need to submit GitHub repos), link to your *hosted prototypes* and *final report* which clearly states theproes and the cones of the different technologies and the proof of the hypoteses.

This project is about developing a proof of concept to support decisions during the planning process of the software development cycle.

- For its purpose, you have to come up with a custom business scenario similar to the one listed at the end of this document.
- Next is to develop 3 hypotheses - these are usually statements that need to be validated during the PoC.
- Once you do a proper research of the market for posible solutions and stacks you will have to narrow it down to developing of prototypes (two for each tier) of 2 stacks and compare the performance and the expenses, along with the different other factors such as eventual learning curves or more assets onboarding or eventual resources upgrade.
- One of the testing types is called a stress test, and you can see more details on the link below https://www.geeksforgeeks.org/how-to-perform-stress-test-on-cpu/
- For front-end testing, you will test the performance with the tools from the browsers, and for the databases, you will need to test the security

Proof of concept definition

What is a proof of concept?

The proof of concept definition is simple: It’s a way of demonstrating the viability of an **idea** or **concept**. Its application extends into the development of technology or software.

In the project management process, you can view the proof of concept as a green light indicating you’re ready to progress into the product development phase.

The idea behind drawing up a POC is to test an idea or concept before you've heavily invested resources into one solution or another. It can help you feel more confident that what you're making or developing will indeed solve the problem or customer/business point you're targeting.

Note: When you have an idea of what a product or service will be like, you can invite feedback from team members, users, and stakeholders that can then be incorporated into your development process.

In any proof of concept, you would have:

- **A clear objective**: Without a clear idea of the purpose of the project, you won't know whether it's capable of solving a particular problem
- **Business specifications**: A general overview of the features and functions you'll test will help guide your POC process
- **A demonstration**: Some form of a working prototype that demonstrates the product or software's capabilities will serve as evidence that you're good to go ahead
- **Evaluation**: Define what success looks like and what data you can collect during the process

What are proofs of concept used for?

There are many use cases for the proof of concept, such as the following:

- **Introducing technology**: Investing in new tech and software solutions can be costly, especially if you make the wrong choice. That's why a POC can prove useful, as you can put the tech to the test before full implementation.
- **Refining processes**: You can use a POC to refine existing processes by trialling new solutions such as automation or AI features.
- **Developing products**: One of the most common use cases for a POC, you streamline the product development process by testing ideas and concepts before entering the manufacturing stage.

Note: When you commit to creating a POC, you safeguard yourself against potential problems by anticipating them in advance. This means you can set a project budget confidently, without fearing that the end product won't serve its intended purpose.

You can also create alignment with broader strategic goals within the company, as you invite input from all parties so everyone with a vested interest in the project can approve before you start work.

How long should a proof of concept be?

The length of a proof of concept will vary according to the team and the project's scope. Since each POC process is unique, there’s no universally agreed-upon time frame for completing one.

If, for example, you’re in an enterprise that wants to trial Amazon Web Services to scale with cloud-based infrastructure, you might look to run a department-wide proof of concept. During this POC process, you would run several **stress tests** and get to grips with the features and functions to see if they align with your strategic goals.

This process could take several hours or days, depending on how thoroughly you test the solution and what you need it for

Stress testing is a technique to test a system's performance under extreme conditions. For example, a website handling a huge number of users at once might be overloaded by mimicking a sudden increase in traffic.

A step-by-step process for putting pen to paper and writing an effective proof of concept for a product or project.

1. **Demonstrate the need for the project or product**

The first step in creating a proof of concept should be to identify the problem. Creating a customer experience strategy will help you pinpoint what direction you need to go in with your project.

If it’s an internal tool to streamline productivity, is there data to suggest there’s an opportunity to improve performance levels?

If it's a customer-facing software solution feature, is there evidence from your market research or customer feedback that implies it would go down well?

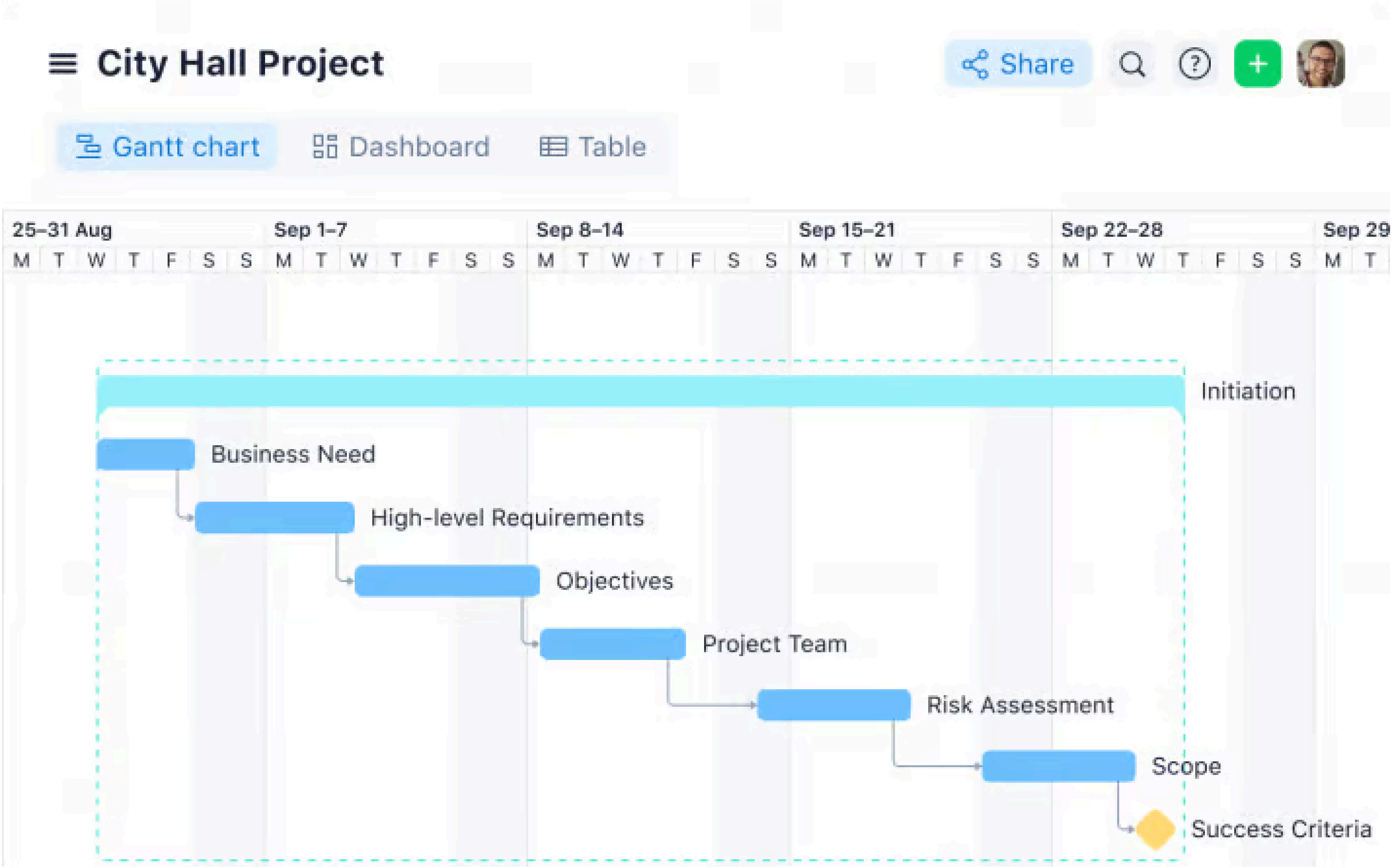
Without some form of evidence to back up the need for the project or product, the proof of concept is irrelevant as you're testing an idea that likely won't land.

2. **Create a roadmap for the solution**

Next, develop a roadmap for the proposed solution that gives you a blueprint for execution.

The proof of concept should just be one small cog in the larger wheel of the product development process, so use this opportunity to plot out the entire roadmap for the product or project.

Detail what purpose it will serve and whose lives it might improve. Include all the technical details such as specifications, functions, and features. Also, map out how you'll bring the product or project to life, be it through project management software, a manufacturing process, or another method.



3. Develop a prototype

Once you have a roadmap in place, you have a starting point.

From here, you can develop a working prototype that tests your theories about the project or product in order to prove feasibility.

The prototype can be a watered-down version of an end product, a small-scale version of a software solution, or anything else that allows you to test your ideas and put them into practice.

4. Test the prototype

When you've built your prototype, you've proved that it can exist in some form. Now you need to stress-test it and put it through its paces to accurately examine viability.

The product development team can carry out thorough testing of the prototype according to the success criteria they developed and constantly refer back to its intended purpose. You can also bring in team members from other departments or stakeholders to get different perspectives and suggestions for changes.

5. Gather results and feedback

In this stage of the POC process, your primary goal is to review the data you've collected and see whether it satisfies your success criteria.

Let's say you work for a coffee machine manufacturer and you're testing a new type of machine that keeps coffee hot for an hour after brewing. With your POC prototype, you could run several experiments with different types of mugs and settings to see if you can hit the sweet spot of remaining hot for 60 minutes after the brewing process is over.

If the coffee were to remain hot after brewing but only for 30 minutes, this would fall short of your initial success criteria. From here, you might decide to revisit the ideation stage or shift your marketing to align with the altered vision for your coffee maker.

6. Present the proof of concept for approval

The final stage of the process will often be presenting your proof of concept for approval.This is where you'll submit your findings to see whether you have the green light to go ahead with developing the product or project.

Provided your POC satisfies the initial success criteria, which were unanimously decided, you should have no issue getting the approval you're looking for. If you use a project management solution such as Wrike, getting approval for a POC document is as simple as passing it on to the right person who can then examine and approve it directly in the software.

Proof of concept examples

Sometimes it helps to see real-life examples to fully understand a process, which is why we've put together these three examples to help you visualize what it could look like for you and your team.

Proof of concept in software development

In software development, one of the biggest advancements in recent years has been machine learning (ML). With ML, software developers can leverage artificial intelligence to automate core tasks or predict behaviours to provide solutions to emerging problems. However, many ML models are around, so it can be difficult to know which might best improve your software development process.

With a POC, you can try out various models to see if they satisfy your success criteria and help you iron out bottlenecks or other issues in the software development pipeline. By feeding the ML model a smaller dataset, you can analyze the results and see how much of an impact it could have on a larger scale.

Project Business Scenario Example!

Software company XYZ, after acquiring a company ABC with their existing clients, inherited a set of applications for Investment and Insurance. The asset/resources budget of ABC is $ 450,000 annually, and there are 7 employees to support the current processes. The management considers upgrading the existing systems by applying automation, AI and microservices architecture. Since it is a required step of the Integration process, decisions in that aspect should be made. While XYZ is relatively up to trend and has the majority part of its services in Azure developed by .NET core (which defines the teams of developers' knowledge stack), the ABC functions on legacy JavaScript vanilla ASP classic and Delphi applets.

This proof of concept has the purpose of aiding decisions against the backdrop of revamping the existing systems and migrating the full stack either on the premises of the company or in the Cloud (Azure, AWS). The main systems are **A.)** for creating customer investment portfolios by investment advisors, and **B.)** Selling and managing various Insurance products and their client/representative processes.

Pick 3 hypotheses to compare and group your analyses in one report.

A **proof of concept (PoC) hypothesis** is a statement about the feasibility or viability of an idea, technology, or product, which is then tested through a PoC exercise to determine its practicality.A new system or web app can improve customer service response times by 20% compared to current methods.

WDE point of examination example

Middleware	**Node Express vs Python**
Client-side	**JavaScript vs Python**
Database	**Mongo vs MS SQL**
Hosting	**Azur (DevOps) vs Jenkins**
IDE	**VS Code vs IntellyJ**
Frameworks	**Next vs Flask**