

Project Plan

ISAAC Sensor Monitor

ISAAC

<<

This template can be used for all projects, especially software engineering projects. Chapters or parts that are not applicable can be removed.

Text in italic is background information and must be removed in the final version of your project plan.

Note that this is a template and can be changed for own purposes, e.g. you can adapt the layout to the layouts as used at the company of your internship.

For your project name, think of a name that highlights the most relevant aspect of your project, and also specify whether it is about graduation internship or third year internship.

>>

Inhoudsopgave

Inhoudsopgave	3
Version history	4
1. Project assignment	7
1.1 Context	7
1.2 Goal of the project	7
1.3 Scope and preconditions	8
1.4 Strategy	9
1.5 Research questions	9
1.6 End products	10
2. Project organisation	11
2.1 Stakeholders and team members	11
2.2 Communication	12
3. Activities and time plan	13
3.1 Phases of the project	13
3.2 Time plan and milestones	13
4. Testing strategy and configuration management	14
4.1 Testing strategy	14
4.2 Test environment and required resources	14
4.3 Configuration management	15
5. Finances and risk	16
5.2 Risk and mitigation	16

Date	:	Date
Version	:	Version
State	:	State
Author	:	Author

Version history

Version	Date	Author(s)	Changes	State
1.0	23-2	Redouan, Najim,Jeremey, Bruce, Rory	Werken aan de eerste drie hoofdstukken van het project plan	finished
2.0	1-3	Redouan, Najim,Jeremey, Bruce, Rory	4, 5.1 & 1.5 gemaakt	in progress

Distribution

Version	Date	Receivers

Contents

- 1. Project assignment 4
 - 1.1 Context 4
 - 1.2 Goal of the project 4
 - 1.3 Scope and preconditions. 4
 - 1.4 Strategy. 4
 - 1.5 Research questions. 4
 - 1.6 End products. 4
- 2. Project Organisation. 6
 - 2.1 Stakeholders and team members. 6
 - 2.2 Communication. 6
- 3. Activities and time plan. 7
 - 3.1 Phases of the project 7
 - 3.2 Time plan and milestones. 7
- 4. Testing strategy and configuration management 8
 - 4.1 Testing strategy. 8
 - 4.2 Test environment and required resources. 8

4.3 Configuration management 8

5. Finances and Risk. 9

5.2 Risk and mitigation. 9

1. Project assignment

1.1 Context

De projectopdracht komt van ISAAC, een IT-software bedrijf die zich vooral richt op digital commerce en de integratie van financial services, een zogenoemde Full-service Digital Agency. Het bedrijf richt zich ook op de E-commerce en Cloudplatformen, zo zijn ze van veel markten thuis en kunnen zij zeer goed helpen met het opzetten of verbeteren van webshops voor/van andere bedrijven en ondernemingen.

Het bedrijf heeft in het kantoorpand een aantal sensoren hangen, deze data slaan ze op voor onderhoud en het behouden van een gezond werkmilieu.

1.2 Goal of the project

Het doel

Het doel van het project is om werknemers inzicht te geven over de temperaturen rondom een kantoorafdeling en om dit op een duidelijke manier weer te geven in een UI. In het kantoor staan op verschillende plekken DHT-12 sensoren opgesteld die constant de temperatuur meten. In totaal zijn er ongeveer 20 sensoren verdeeld over het kantoor die ook verbonden zijn aan verschillende microcontrollers. De data die de sensoren vergaren wordt via het MQTT protocol verzonden naar een server. In de huidige situatie wordt de data gebruikt voor het weergeven van een heatmap. Wanneer de conditie rondom temperaturen in het kantoor verslechterd, krijgt de kantoomanager hier een melding van.

Hoe ziet het verwachte eindresultaat eruit?

Bij het afronden van dit project moet er een infrastructuur staan die de data van de sensoren gebruikt om deze te weergeven in een informatieve UI. De UI is een web-based applicatie die het product wat het bedrijf op dit moment gebruikt moet uitbreiden met bepaalde “smart solutions”, zoals;

- Een beter Heat Map Display met animaties dat inzicht geeft van de temperaturen in het kantoor in een werkweek.
- Mogelijkheid om Sensor Quality Reports uit te voeren om de kwaliteit van de DHT-12 sensoren in de gaten te kunnen houden.
- Een manier om de weersverwachting voor Eindhoven te weergeven met een voorspellings-feature voor de werkconditie binnen het kantoor.

Wat zijn de voordelen van dit project?

Gebruikers kunnen op een overzichtelijke manier zien wat de huidige temperatuur is binnen het kantoor en of de werkomstandigheden ideaal zijn om in te werken.

Op wat voor manier(en) is het project waardevol voor het bedrijf?

Dit project moet ervoor zorgen dat het bedrijf een beter inzicht krijgt over de situatie binnen het kantoor door middel van een gedetailleerde UI. De UI beschikt over extra features die het mogelijk maken om voorspellingen doen. De kantoormanager kan hier dan actief op reageren wanneer de werkconditie in het kantoor verslechterd.

1.3 Scope and preconditions

<<What activities, and which endproducts (to what extent or quality) belong to the project, and which don't >>

Inside scope:	Outside scope:
1 MQTT	1 Sensor data
2 Visual Studio .NET	2 Plattegrond & voorbeelden
3 React	
4 Postman	
5 Azure DevOps	
6 Google Docs	
7 Docker (Desktop)	

<< Indicate any preconditions. E.g., think of technology choices that have already been made by the company. Note that you are also expected to retain a critical, but constructive, mindset for choices already made >>

1.4 Strategy

<< Describe the strategy for your project (the approach). E.g., waterfall, or an agile approach like scrum, and justify the choice. >>.

Onze strategie voor de aanpak en management van het project is Azure DevOps. Wij gebruiken hier de agile S.C.R.U.M. methode. Na het vergelijken van GitLab en DevOps zijn we tot de conclusie gekomen dat DevOps een professionelere aanpak voor ons aanbiedt.

Uiteindelijk hebben wij S.C.R.U.M. gekozen, omdat deze agile manier van project managen veel gebruikt wordt in professionele omgevingen en deze ook een continu juist en duidelijk overzicht geeft over wat er nog moet gebeuren om tot een goed eindproduct te komen. S.C.R.U.M. is een flexibele software ontwikkelmethode, waarbij korte sprints worden gebruikt om te evalueren. Zo houdt je de werkdruk hoog en is iedereen open over zijn of haar skills en vooruitgang.

1.5 Research questions

<<

Describe the research questions that are most relevant to your project. For each research question, describe the approach and/or methodology. Use the Dot Framework to specify strategies and methods - see <http://www.ictresearchmethods.nl> for details.

Note that research is not only part of the initial phases (like analysis) of the project, but runs throughout the whole project. E.g., in the realization phases, you will probably do research in the Workshop and Lab context.

Also realize that during the project your research questions may change, and that new ones will come up. That normal for any project J, and is not a problem as long as you involve the right stakeholders, and keep your deliverables updated and in sync.

>>

Onderzoeksvragen

1. Wat voor architectuur willen wij gaan gebruiken?
 2. Wat gebruiken wij voor de communicatie tussen applicaties?
 3. Hoe ziet onze CI / CD pipeline eruit?
 4. Welke server gaat de applicatie op runnen?
 5. Wat voor data krijgen we geleverd?
-
1. Op welke manier gaan wij onze applicatie opzetten? We zullen hiervoor allerlei verschillende manieren van architectuur gaan moeten zoeken. Doen we alles in 1 grote project? Of splitsen we de taken op? Hierbij raadplegen we hulp van het internet en eventueel docenten. Dit is dus stakeholder analysis.
 2. Het onderzoeken van waarmee we onze microservices willen laten communiceren. Is er 1 API die met meerdere API's communiceert? Welke tooling gebruiken we

hiervoor? We kunnen eventueel bestaande projecten bekijken die al deze architectuur vorm hebben. Dit is “community research” & “available product analysis”.

3. Welke tooling gaan we gebruiken om onze project te managen. Opties zijn GitLab, GitHub en Azure DevOps. Om tot een conclusie te komen zullen we dit moeten bespreken met de PO's en leden van onze groep. Hierin komt “good and bad practices” in voor.
4. Waar wilt ISAAC het product in geleverd krijgen. Wat zijn de requirements? We moeten onderzoek doen naar wat een MQTT server is aangezien dat in de requirements stond. Explore your requirements zijn essentiële methodes hier.
5. Waarmee moeten we onze applicatie vullen? Wat zijn de requirements? Moeten we eigen data gebruiken of krijgen we data van de klant? Hiervoor zullen we met ISAAC moeten spreken as to what ze nodig hebben in de applicatie. Zo kunnen we uiteindelijk kiezen wat voor data we willen gebruiken in de applicatie. Ook hier komt “stakeholder analysis” & “explore your requirements” in voor.

1.6 End products

<< A Product Breakdown Structure (PBS) lists the end products that you realize, including a description of each product. In software engineering, the end products are more than just the project plan and the application itself. E.g., requirements documents, architecture documents, research reports and test reports are all end products. These are all important products that are required for effective handover. They are also necessary for further maintenance and follow up-projects. The PBS can change during the course of the project..>>

2. Project organisation

2.1 Stakeholders and team members

<<Indicate all stakeholders and team members for your project. For each stakeholder indicate the role for your project. Note that the role that a person has for your project is different from the function the person has. E.g., someone with the function "department manager of department X" can have the role of product owner for your project.

Name	Abbreviation	Role and functions	Availability
<i>Jeremey van der Hansz</i> 06 42269529 Jeremeymain@gmail.com 393712@student.fontys.nl	<i>Jer</i>	<i>Scrum Master.</i>	<i>Maandag : 10-4</i> <i>Dinsdag : 10-4</i>
<i>Bart Delvaux</i>	<i>Delvaux</i>	<i>Product Owner</i>	<i>Maandag: 10 - 4</i> <i>Dinsdag: 10 - 4</i>
<i>Ronald van der Horts</i>	<i>Horts</i>	<i>Product Owner</i>	<i>Maandag: 10 - 4</i> <i>Dinsdag: 10 - 4</i>
<i>Rory van den Akker</i> 436507@student.fontys.nl	<i>Rory</i>	<i>Developer</i>	<i>Maandag: 10 - 4</i> <i>Dinsdag: 10 - 4</i>
<i>Samuil Angelov</i>	<i>Angelov</i>	<i>Begeleider/Leraar</i>	<i>Maandag: 10 - 4</i> <i>Woensdag: 10 - 4</i> <i>Donderdag: 10 - 4</i>

<i>Bruce van de Ven</i> 430690@gmail.com	<i>Bruce</i>	<i>Developer</i>	<i>Maandag: 10 - 4</i> <i>Dinsdag: 10 - 4</i>
<i>Redouan Sammour</i> 0644943344 430575@student.fontys.nl testgebruiker2019@gmail.com	<i>Ridi</i>	<i>Developer</i>	<i>Maandag: 10 - 4</i> <i>Dinsdag: 10 - 4</i>
<i>Najim Doudouh</i> 0685216106 doudouh124@gmail.com 429708@student.fontys.nl	<i>Najim</i>	<i>Developer</i>	<i>Maandag: 10 - 4</i> <i>Dinsdag: 10 - 4</i>

2.2 Communication

<< Indicate the meetings and other channels of communication that you have established, or that you use for your project. Think of communication with all stakeholders including company supervisor, teachers, etc.

In which manner does each communication take place? Think of the goals, the location (or whether it should be online), the timing and frequency, and the attendee list.>>

De manieren waarop wij als groep zullen communiceren zijn : Mail, Discord, Teams en wanneer we weer op locatie kunnen in persoon. We zullen binnen onze groep vooral via discord en op school communiceren. Dit gebeurt elke maandag en dinsdag omdat, dit voor ons project is ingedeeld, Met de project owners hebben we elke 4 weken onze sprint retrospective. Deze zullen we wanneer we hulp nodig hebben contacteren via discord of mail wanneer ze niet reageren via discord.

Met samuel zullen we wat vaker praten. Elke maandag zien we hebben dus kunnen we praten over onze progressie en krijgen we advies van samuel.

3. Activities and time plan

3.1 Phases of the project

Zo we beginnen met een probleem analyse. Dit betekent dat we bekijken want, onze klant eigenlijk echt gedaan wil hebben. Wanneer we dit hebben geanalyseerd kunnen we verder met bedenken hoe we deze problemen het best kunnen oplossen.

Wanneer we weten hoe en wat, kunnen we dit implementeren in onze applicatie. Hierop kunnen we feedback op krijgen om veranderingen te maken later. Dit is de implementatiefase

De project owners zullen onze project evalueren en hierop feedback geven die weer gebruikt kan worden voor het verbeteren van het product. Dit is onze evaluatiefase.

Dan reflecteren we hierop als teams en als individuen zodat we op een betere manier kunnen werken en de implementatie kunnen verwerken in het eindproduct. Dit is de reflectiefase

Als laatste geven ons finale product aan de project owner met alle implementaties erin. Dit is onze handover.

3.2 Time plan and milestones

<< For a waterfall project you can indicate the phases and milestones below (can be adapted as required).

For an agile project describe how the artefacts are planned. E.g., length of sprint (with justification), organization of stand up, demo, retrospective.

>>

Sprint lengte	Stand Up Tijd	Demo	Retrospective
4 Weken	Ma-Di : 10 uur	Laatste maandag van sprint	Na de presentatie

De sprint lengte van onze sprints zijn over 4 weken. De reden hiervoor is omdat, we denken dat we binnen 4 weken weinig resultaat kunnen brengen naar de project owner. Deze extra week geeft ons drie weken met daadwerkelijk werk en 1 week afronding en voorbereiding wat bij fijner vinden.

Onze stand up begint om 10 uur. Dit komt voor mensen die iets later op willen staan of wanneer we een korte presentatie van onze leraar krijgen in de ochtend.

De demo geven we de laatste maandag na van de sprint. Hier geven we een korte presentatie van wat we zouden bereiken deze sprint. Dan een kort demo van de applicatie waar we deze verschillende user stories in laten zien. Als laatste vertellen de project owners over onze doelen van volgende sprint.

We zullen ons hierna op reflecteren. We zullen ons als individu reflecteren en als groep. Zodat we de samenwerking kunnen verbeteren.

4. Testing strategy and configuration management

4.1 Testing strategy

Unit testing - De unit-test zal een geautomatiseerde test zijn. Unit test maakt het mogelijk om de kleinste onderdelen van onze applicatie te testen. Dit zorgt ervoor dat de kleine onderdelen van onze applicatie zeker werken. We willen een code dekking van ongeveer 40 ~ 50%. Dit komt doordat we nog niet veel ervaring hebben met testen. Dit percentage zou voldoende moeten zijn om de belangrijke code te testen.

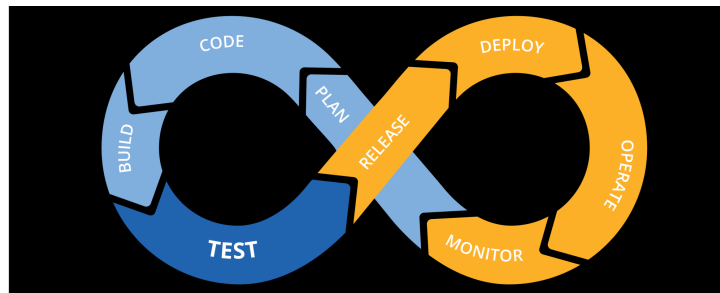
Integration Testing - De integratietest wordt geautomatiseerd. We zullen Cypress gebruiken om de integratietests te automatiseren. Integration tests catch some bigger test gaps which the unit tests aren't able to catch.

User UI Testing - The user UI Testing is niet geautomatiseerd. We gebruiken deze test omdat, het belangrijk is voor de user om makkelijk te temperatuur te checken. Wat de applicatie doet is heel simpel dus, moeten we ervoor zorgen dat de UI duidelijk deze simpele informatie doorgeeft.

Acceptance Testing - De acceptatietest zijn niet geautomatiseerd. De test zijn er zodat we zeker zijn dat alle requirements van de product owner in de applicatie zijn geïmplementeerd en om te kijken of er potentiële veranderingen zijn in de requirements die hun in de applicatie willen hebben.

Voor code coverage testing zullen we Sonarqube gebruiken. Hier kunnen we ook de rating voor security en andere elementen zien. Dit zal ervoor zorgen dat we makkelijker onze software kunnen verbeteren

4.2 Test environment and required resources



CI / CD Tooling:

De tooling die voor CI / CD wordt gebruikt is Azure DevOps. DevOps beschikt over veel features die dit proces makkelijker moet kunnen maken. Zo zijn er al vooraf ingestelde pipelines waar het proces voor bouwen en testen al gedefinieerd is. Het is ook zeer uitbreidbaar, en indien nodig is het mogelijk om een pipeline aan te passen.

Ook kunnen we DevOps gebruiken voor Code Quality Analysis en code scan tooling. Een bekende tool is SonarQube, die constant code quality checks uitvoert op de repository. De tool kan bugs, code smells, en eventueel beveiligingsproblemen vroegtijdig detecteren.

Testing concepts die worden gebruikt tijdens de ontwikkeling:

1. Unit testing & integration testing

Voor het functioneel testen van de gehele applicatie maken wij gebruik van unit- en integration testing. Wij gebruiken unit testing voor het specifiek testen van verschillende componenten in de applicatie, en integration testing om alles te testen als een compleet geheel.

De tools die wij hiervoor willen gebruiken zijn de standaard .NET tools. Voor het functioneel testen wij MSTest of XUnit gebruiken.

2. User interface testing

Met user interface testing kijken we of de UI aansluit met de requirements die opgesteld zijn door de product owners. Het testen van de UI moet uiteindelijk bijdragen aan het de ontwikkeling van een een gebruiksvriendelijke applicatie die makkelijk navigeerbaar is en er goed uitziet.

3. Acceptance testing

De product owners zullen deel uitmaken van acceptance testing om te controleren of de applicatie voldoet aan alle requirements. Als de product owners iets niet goed vinden, of verbeterpunten hebben, dan wordt dat tijdens deze test duidelijker. Dit zal meegenomen worden in het ontwikkelingsproces. Voor acceptance testing zullen we regelmatig een meeting inplannen om de progressie te bespreken en om werk te laten zien. Dit draagt allemaal bij aan de Quality Assurance van het project.

4.3 Configuration management

<< Describe the project approach with respect to version management. This might include things like tooling, branching strategy, promotion-, release- and baseline strategy.

We zullen Azure devops gebruiken voor versiebeheer. We gaan verschillende branches maken van onderdelen van de applicatie. Deze branches zullen verschillende taken zijn op ons scrumbord. Deze branches kunnen altijd gepushed worden. Maar ze kunnen alleen gemerged worden wanneer er geen errors aanwezig zijn binnen in de branche. There should be as much commits as possible within the branch, Every essential implementation to achieve it should be committed in the branch

The baseline will be a couple user stories which will be make a baseline. This baseline will exist out of a couple branches. There will be four baselines for us because we have four sprints. Which will each contain their own stories/tasks

Also, when relevant, think of a mechanism to deal with change requests and problem reports.>>

5. Finances and risk

5.1 Risk and mitigation

<< Investigate and define all risks affecting the project. For each risk indicate what has been done, or will be done during the project, to prevent the risk from being actualized, and define the mitigation actions, such as what you plan to do if the risk actually eventuates.

In a more elaborate version, you can also label the risks with their chance of occurrence and impact. The advice is to focus on risks that have both a real chance of eventuating and some considerable impact. Direct risks, like what to do if your company supervisor is not available anymore, should always be described, as they have happened in the past quite regularly.

>>

Risk	Prevention activities	Mitigation activities
1 Er is een tool die een te hoge learning curve en daardoor niet goed geïmplementeerd kan worden.	Doe genoeg onderzoek naar de tool en vergelijk ook met andere tools.	Heb een alternatieve tool klaarstaan.
2 1 of meer personen worden ziek of zijn verhinderd.	Duidelijk beeld creëren van wat iedereens taak is binnen het project.	De taak kan overgenomen of eventueel uitgesloten worden.
3 Er is niet genoeg tijd om een goed eindproduct neer te zetten.	Hou in de gaten of er niet te lang wordt gewerkt aan specifieke taken.	Rond de te lang durende taak af zodat er geen achterstand komt.
4 Verhinderingen of onverwachte uitloop worden niet goed doorgegeven, dit kan tijd kosten.	Communiqueer elke projectdag alle onverwachte obstakels door.	Los het probleem met de groep op of schrap een te ingewikkeld onderdeel, zodat het project process niet gehinderd wordt.

5 Er zijn onverwachte complicaties in de wereld die invloed hebben op het project, bijvoorbeeld een pandemie, natuurramp of oorlogssituatie, maar ook een computercrash.	Heb overal een back-up/cloud opslag van, zorg dat iedereen thuis een werkplek heeft en weet van een goede communicatie tool zoals Teams of Discord.	Recover de verloren data van een back-up point/cloud opslag als gevolg van een computercrash.
--	---	---