University of Southeastern Norway

Systems Engineering

Home Exam

May 29, 2020

Contents

Chapter 1

Thinking patterns in systems engineering

1.1 Thinking like a systems engineer

Systems engineering is applied when we are working with large complex systems. For that purpose, it important to understand how a systems engineer goes about solving the problem in terms of what the different thinking patterns they follow. These patterns are more connected to the design part of the development and we quickly realize why applying systems thinking is paramount. There have been many Systems Engineering and design failures due to a lack of applying Systems Thinking like the Tacoma Narrows Bridge collapse in Washington 1938-1940.

Figure 1: The collapse of Tacoma Narrows Bridge in Washington 1940

The area was known for strong, and on a particular day with the a 42 mph wind, it resulted in the bridge collapsing. From a design perspective we understand that some failures were not being able to adequately address interactions between system components and the environment. Also, not being able to identify relevant environmental factors such as wind, rain and temperature which we know now a great tool for this is a system context diagram. This one of several unfortunate examples where see the lack of systems thinking. Luckily, we have come a long way since then, and there are now several thinking patterns for systems engineers to follow in the development efforts, and following I will look at 3 of them.

1.2 Decomposition/composition thinking

A famous Stanford Professor once said: "If you can't solve a problem, then there is an easier problem that you can solve: find it". This is what comes to mind when I think about decomposition/composition thinking. It's all about dividing our system into subsystem, it allows us to unfold the complexity of our system and provides a better understanding and detail for each subsystem. Big complex problems are comprised of smaller and more easily solved sub-problems or tasks, so by logically identifying these smaller problems and determining how to use the combined solutions to solve the bigger problems we perform this kind of thinking. The challenge with this thinking however, as the book mentions, is how do we integrate all the parts into a well-functioning system while keeping the integrity of the system. For that purpose, it is important that we allocate top-level functions to subsystems and create interfaces between these subsystems, so you can step to other levels to investigate the functions that are needed to accomplish the top-level functions.

This thinking pattern proved to be quite helpful in handling the complexity of our system. Considering how comprehensive an energy system can be, decomposing it into sub-system helped us tremendously in understanding what we were dealing with and the different levels of functionality. This was true especially when we did our functional analysis. Firstly, we looked at the functional architecture, and by using the decomposition/composition thinking we made this model. It shows the architecture of our system on a level basis and by connecting top-level component. with the sub-level component. Since we are creating an energy-system, we took the view that since we are creating an energy system all other parts of our system will be a sub system to this. This is helpful both for us as designers of the system, and for the customer as it unfolds the complexity and shows how all elements of our system working together.

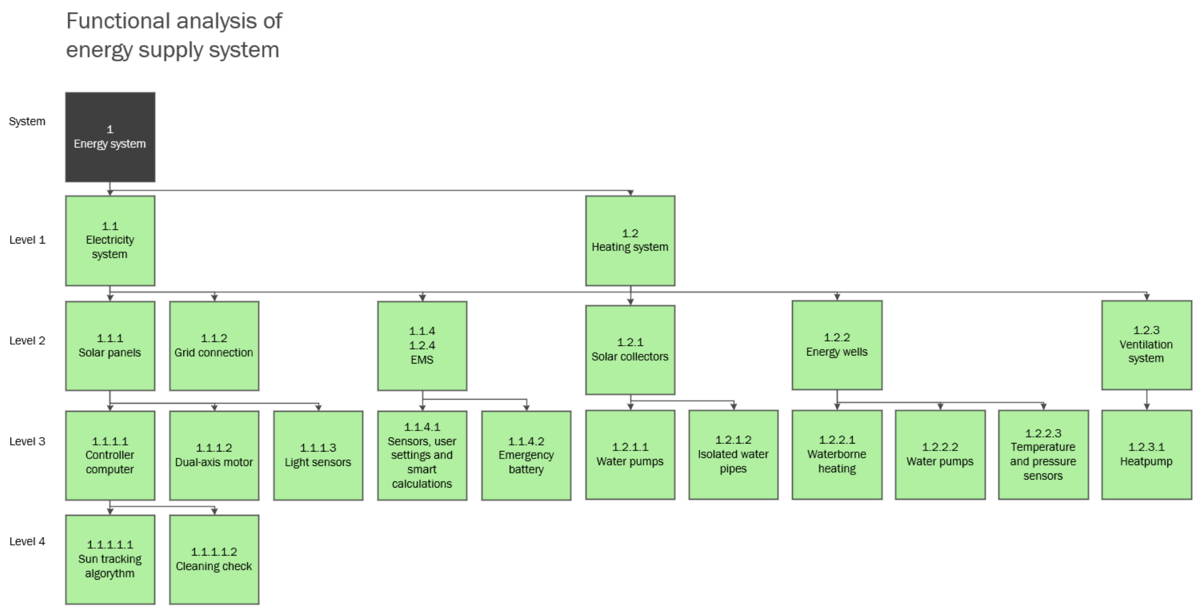


Figure 2: Functional architecture diagram

We struggled with the design of the energy system initially because we didn't know how all the different components would be connected. But as the book suggests, and we later learned after trying and failing, it is important to create interfaces between subsystems, and that is what we did with the functional architectural diagram where we connected the components to see that the solar panel system will have dual axis motors and that the energy management system will have a backup battery.

By applying this thinking pattern, it allows to focus more on a specific sub-system. From the higher-level functionalities, we can look at one functionality that we decomposed further. Doing this helped us define the functionality and be able to trace it throughout our system.

1.3 Hierarchical thinking

In hierarchical thinking, the system designer has to consider ranking authority, facilities and priorities of the system's parts. In the ranking system the element at the top has the highest priority, and the other elements follow in order of what priority is next highest. We can use this sort of thinking when he have sub-systems or elements that are dependent on each other or have some sort of connection.

An example of how we used this sort of thinking pattern is regarding how we “charged our wells”, in other words how made sure that the water in our vertical wells remained at a stable temperature throughout the year.

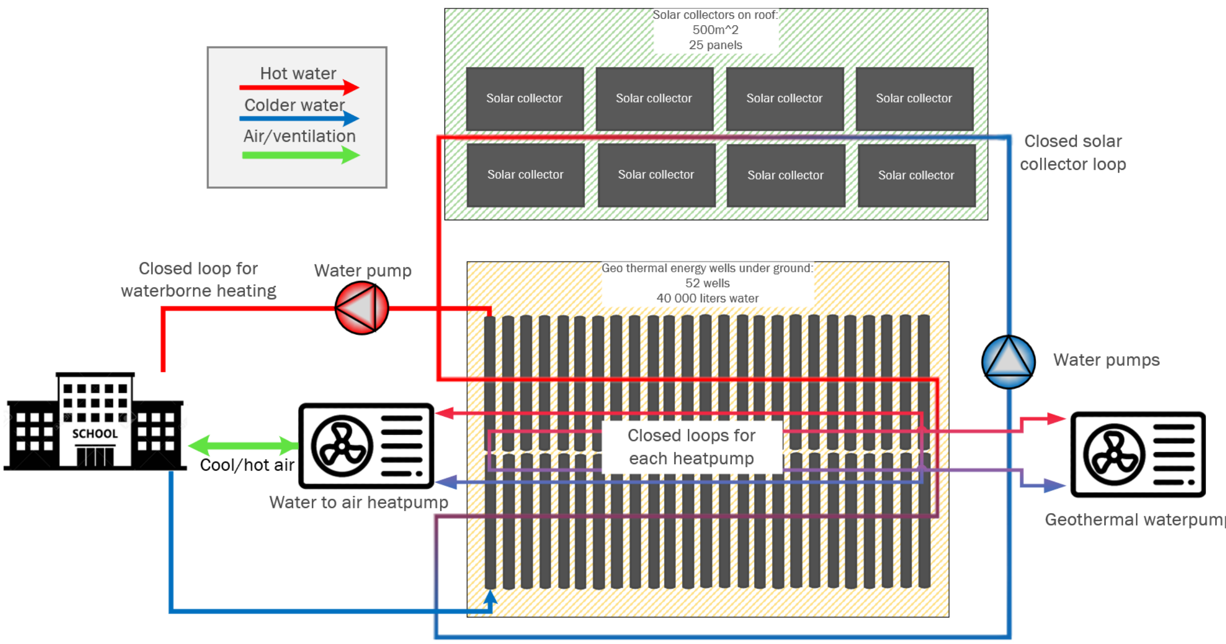


Figure 3: System diagram showcasing the well structure

It is a requirement that the liquid within the well should remain at a stable temperature throughout the year. For that purpose, we needed to design a system where that would be the case. By using hierarchical thinking, we made a system where the solar collectors would ideally supply the heat for the water, but since it can’t maintain the level of production needed throughout the year, we implemented the use of heatpumps. A water-air CO2 heatpump that use excess air from the building and outside climate, and another water-water heatpump. These will “kick-in” when the solar collectors, are not producing at the required levels. In this case, the solar collectors and the solar energy has the highest rank, and we see that form the figure which showcases a pyramid like form which is common for hierarchical structures.

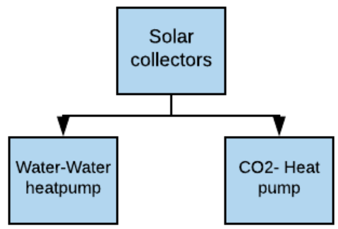


Figure 4: Hierarchical structure for charging the wells

1.4 Dynamic thinking

The systems we design rarely stays the same over time. It interacts with the environment and is often dynamic. Looking at the system from a dynamic perspective is therefore essential and here we look at how the system changes over time, how the environment changes over the time and what happens when a change in input or output occurs.

We simply could not avoid doing dynamic thinking when we set out to design the energy system for a school, because the energy system has to change with the environment, so we knew we had to make an adaptable energy system, and for that we used dynamic thinking. The dynamic thinking also overlapped the operational thinking because we also had to look at how the system would act in a real-life environment.

We had to deisgn the system in a way that where we could take advantage of the solar energy. For that we implemented solar panels and solar collectors, but when the weather changes we can't simply rely on the solar energy for electricity, so we knew that we had to design a grid-tied system where we had the possibility to buy power during harsh winter times and where we could sell excess energy in the summer times. In that way, our system would be adaptable to the environment.

We also did a lot of research on how our system components would change with time and after researching our technology and hardware, we came to the conclusion that our system would not change much with the time, mainly because we have an automated system and the life cycle of our main components such as solar panels and collectors, inverter, pipes and heat-pump are expected to have a life-cycle 30 years.

Chapter 2

Backward and forward traceability in requirement management

2.1 What is traceability in requirement management?

When designing big complex systems, we must have traceability models and traceability aids should be in place because in these systems we have a quite complex web of relationships. For that purpose, we need to make sure that we can trace our requirements, so we know if they have been adequately considered during each phase of the project and that there aren't any inconsistencies.

It all starts with a concept and our customers. This is where the requirements gathering process begins. After analyzing our stakeholders and customers we can start the process by getting all the different requirements they might have. We look into their concerns and translate those into requirements. By doing this identification process the right way, the requirements will express what our stakeholders and customers wants from the system.

Another important type of requirement is regarding what we as system designers want from our system, the system requirements. The focus here should be to develop requirements for the system that are fulfilling a concurrent issue, requirements that are realistic, testable, unambiguous and able to make the product attractive in the current market. These can be divided into functional and non-functional. The functional requirement is describing the behavior of the system as it relates to the system's functionality while the non-functional requirement elaborates a performance characteristic of the system.

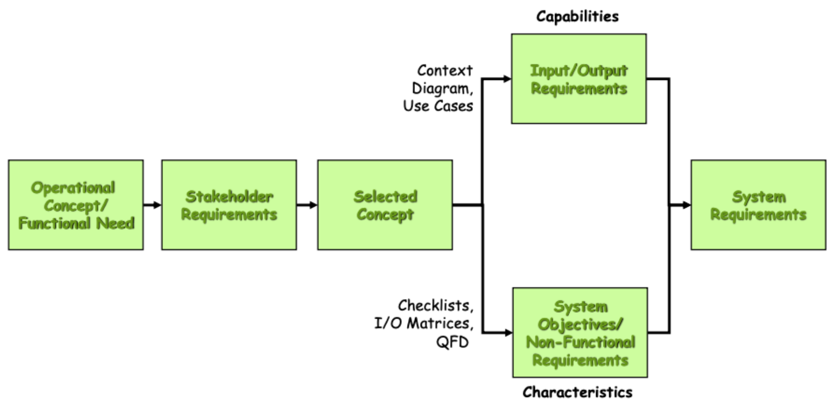


Figure 5: Requirements development method

This is where traceability comes into play in requirement management. From our perspective, as designers, but also for our customers the idea of tracing a requirement from when it is collected to when it is fulfilled is vital. Not only does it provide a better understanding for both parties, but for us as designers, it makes sure that we have considered the requirements in each phase and that there are no missing parts.

With systems engineering applied to big, complex projects, there are many different people with different backgrounds working on the same project. So, in order to trace the requirement during its life-cycle we need traceability. It is very common practice that a stakeholder can change his/her requirement leading to lots of time and money being spent because we didn't trace it properly and now much work has to be done to change that. But, if we implement traceability in our development efforts, we can track that change of heart from a stakeholder all the way from the start leading to more efficient methods.

We can do this traceability both forwards and backwards. That is, tracking a requirement from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases. In forward traceability we trace a requirement to components of a design or implementation. In backwards traceability we trace the ability to trace a requirement to its source, i.e. to a customer or stakeholder. By forward and backwards traceability, we can trace a requirement from where it began and by who all the way till where it was implemented.

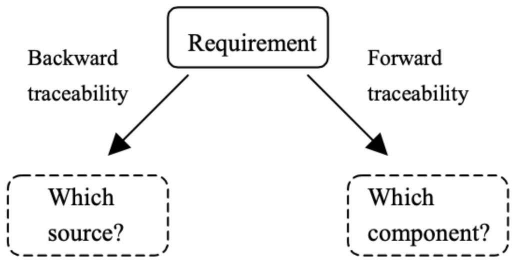


Figure 6: Forward and backwards traceability

2.2 Our work with traceability throughout the system development process

I was the one in our group with the responsibility of grouping and documenting the different types of requirements. So, when I started with this task, I knew that traceability would be important for every requirement, especially the requirements which were crucial for our system, and I needed to understand the process of how we would know whether that requirement would be fulfilled or not. For this purpose, I made a requirements table ranking their priority based on high, medium and low order of importance. High priority means that the requirement is absolutely critical for the system, and without it the system wouldn't be able to perform to its full capacity. Medium priority indicates that the efficiency of the application would be greatly improved by this requirement, but it's not critical for the system. Finally, low priority means that the feature would simply be nice to have, and it may consist of cosmetic changes or non-critical inquiries.

ID	Requirements	High	Med	Low
1	The Energy Management System should show information about the system's components, energy production and efficiency Infrequent service/maintenance	*x*		
2	The systems should require infrequent service/maintenance		*x*	
3	The system should include a digital display at the entrance of the building that shows live selected data			*x*

Figure 7: Requirements table

Ranking the requirements as shown in figure 7 helped us identify the requirements with the highest priority, the ones that are critical for the system of interest, and make sure that those are met firstly. Every requirement has its own ID making it easy to track. This made the implementation part of the development easier too because by using forward and backwards traceability we had the option of seeing which component and which source led to which requirement giving us a top-down or down-up view. There were also many instances where we had to add, delete or change a requirement and having this setup made that process easier throughout our development process.

2.3 Tracing a stakeholder requirement top-down

We start by communicating with our customers and stakeholders and we gather all their concerns and what they want from our system. Capturing the stakeholder's and customer's concerns is critical throughout the entire development process. If we don't succeed at that, we have created a solution in search of a problem, not the other way around because then who is our system for? So, the first thing we must do is always understand the problem so that we can solve the right problem, the problem that the customer wants to solve.

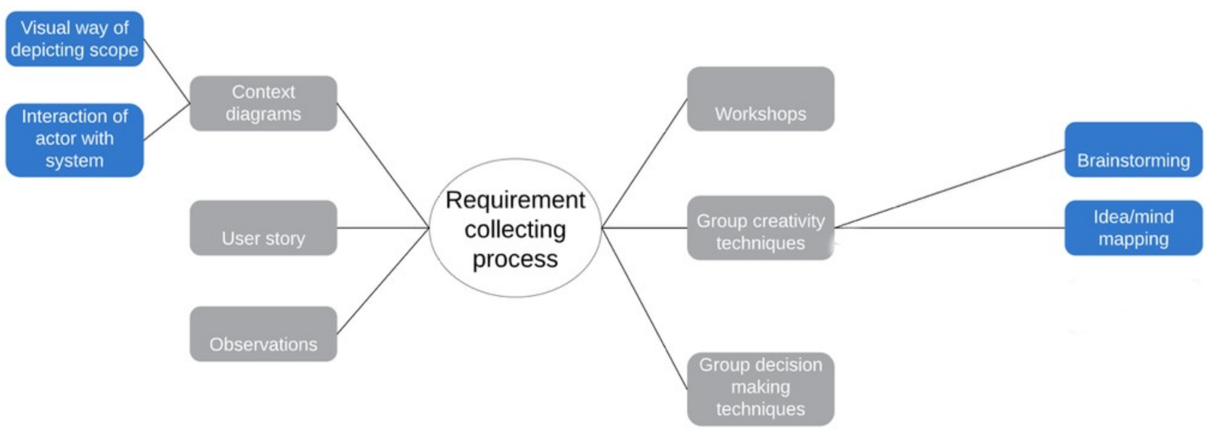


Figure 8: Our customer requirement collecting process

From the collecting process we found a stakeholder requirement with a unique identification and a high priority which says:

ID	Requirements	High	Med	Low
1	The system should be safe to use	*x*		

Figure 9: The selected stakeholder requirement

If we unfold this requirement from the stakeholder's perspective, good safety means reduced cost in terms of service and no unfortunate incidents. Stakeholders that share this requirement in our case are including but not limited to Oslo municipality, the government, service personnel and school administration. Translating it to a systems perspective good safety will contribute to our system being durable.

Now that we have identified the stakeholder requirement and understood what it entails, we can move forward to the next development phase, the design. The requirement tells us that we have to design the system in a safe manner. For the energy system this means that we must ensure proper insulation and not have any hazardous materials, it also means that we must be able to monitor the water pressure and temperature in our wells, the status on our battery and that we must have smart systems to detect any leakage.

After the design phase we perform technical research where we compare different technologies to use for our energy system. We also perform trade-off study where we look at pros and cons for different components and technologies regarding our system in order to select the ones that are best suited for our system given the requirements. After this research, we can move to the implementation phase.

Most notably, we choose to implement an automated system, meaning from the energy is "produced" til it's distributed the whole process will be automated, resulting in less physical interaction with the system and minimal maintenance efforts. More specifically, we choose the right materials and liquids, the specific smart functionalities and monitoring systems we want to implement in order to ensure good safety. To prevent any hazardous situation, we will implement a liquid solution with water and propylene-glycol and for the piping material we will select a plastic called polyethylene which can withstand tremendous amounts of pressure and is widely used in the market for its reliability and endurance. As for the component surveillance, it will be connected to our energy management system which the user can access through a user interface. These are all examples of how we operate in the implementation phase in order to fulfill the requirement.

As for verification and validation, the components and technologies selected will be tested individually and then in their respective sub-systems, but proper validation can only be done once the system is installed and deployed.

By tracing this requirement “top-down” from identifying the requirement through our development phase and finally implementing the requirement into a solution, we see the advantage of traceability in requirement management. This way we can now do forwards traceability and see what component the requirement relates too (i.e battery), but we can also do backwards traceability and see what the source of the requirement was (i.e Oslo municipality). Having the forward and backwards traceability eliminates any inconsistencies and helps us make sure that the requirement has been adequately considered during each phase of the project.

Chapter 3

Identifying and managing interfaces

3.1 Boundary and scope

Before we can talk about identifying and managing interfaces it important to mention boundaries, scope and the system context. This is what tells us what we are designing, what we are focusing on and what surrounds our system.

In our case, one of the first things we did collectively was to define the system boundaries. Considering our case is comprehensive in its nature, defining system boundaries was a vital part in the design process. We set boundaries for our system and subsystems, so when we do any work, we make sure not to overstep those boundaries which in turn saved us much time. An example for this is in the early stages we had to make a decision of whether we wanted the design responsibility of the building as well as the energy system or mainly focus on the energy system alone, and work alongside a construction company.

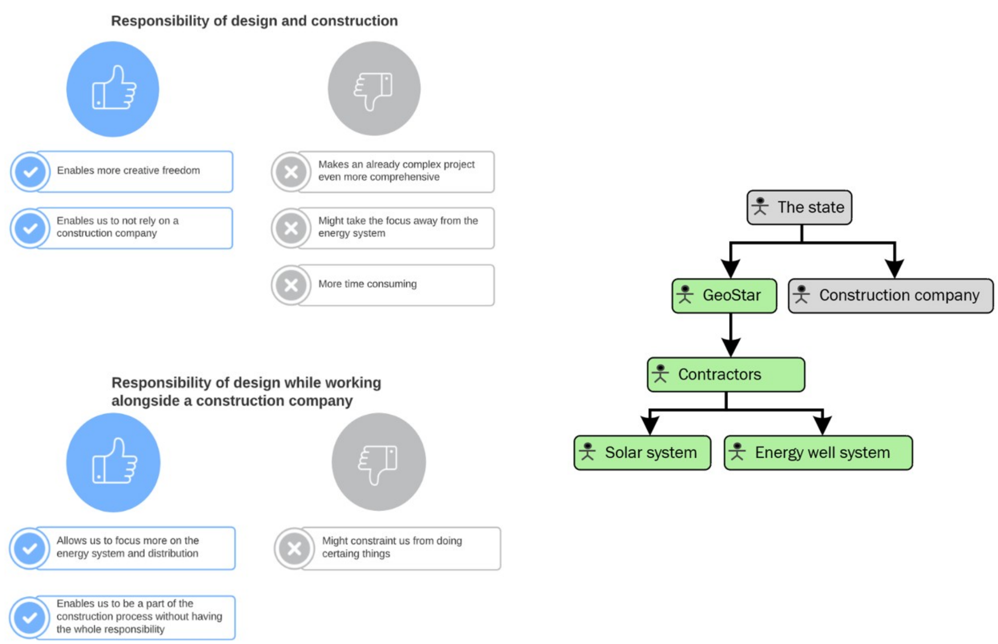


Figure 10: Our role

From the figure and several discussions, we had, we deduced that working alongside a construction company would be the best idea. The main reason was that it would make designing the energy system our top priority and main focus. We believe that, by working alongside a construction company it would allow us to have a say in the process of selecting materials, transportation and installation in order to make sure that the work being done does not leave a massive carbon footprint. This allowed us to narrow our sights into what our design responsibility really is and help set the boundaries for our system.

3.2 Internal and external interfaces

The practice of interface management begins at design and continues through operations and maintenance. Interfaces are the functional and physical connections at the boundaries of systems that are designed to inter-operate with other systems. There are many types of interfaces, including communications interfaces, data interfaces and hardware interfaces.

It is important for us to differentiate between internal and external interfaces because it influences how we can prepare for problems and how much we can influence the architecture of the system over time to make it more robust. With internal interfaces we control both endpoints of the connection. If something goes wrong, we can look at all involved components and the infrastructure and decide how to fix it. External interfaces however, limit our influence on one part of the connection as they are connections to external factors such as the weather or the internet.

The context diagram is a great way of depicting the environment in which our system exists. By depicting the project scope at a high level of abstraction, the diagram doesn't reveal anything about the system internally or its functionality, but it clearly shows every entity that interacts with the different interfaces in our system. For our energy system and that it would directly affect our energy production because we heavily invested in solar energy.

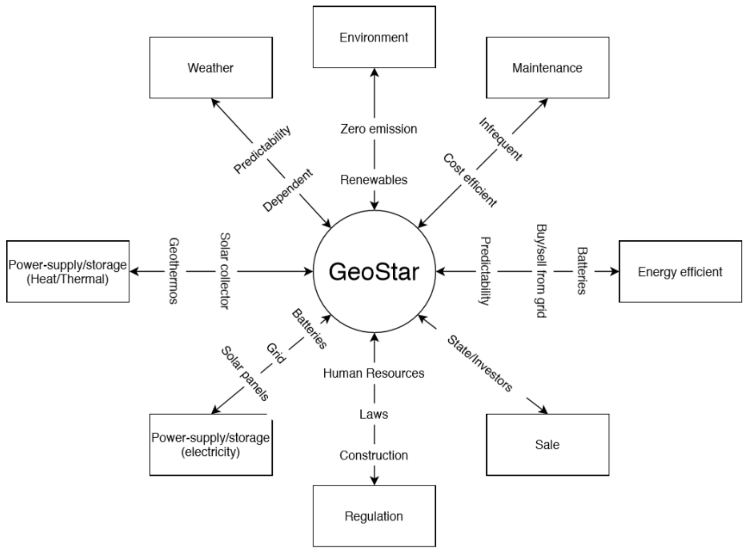


Figure 11: System context diagram showcasing the different interfaces

Another way in which we identified and managed the different interfaces was with our "entire system diagram". We invested good amount of time developing this model because we knew that making this in a detailed manner would provide us with many benefits. One of those benefits is that it shows the different interfaces which made it easier for us when we wanted to implement new sub-systems because then we knew which external or internal interface it would have a relationship to.

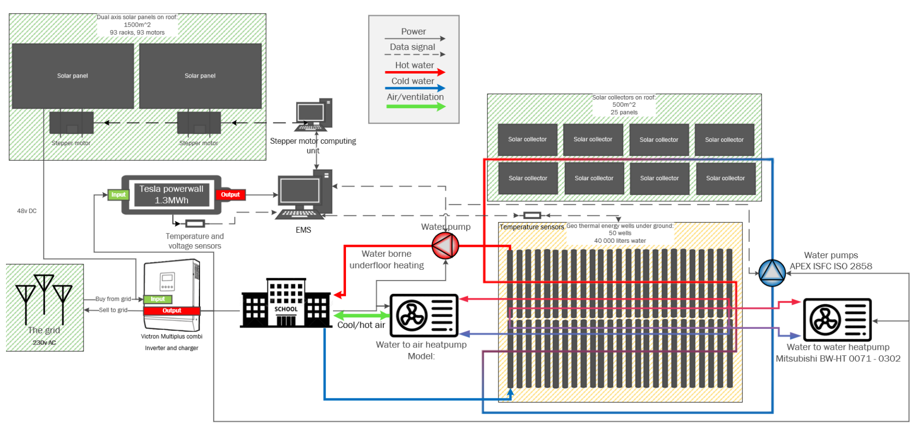


Figure 12: Entire system diagram

Chapter 4

Architectural views

4.1 Functional, operational and physical architectural views

A system can have three architectural views, functional operational and physical. Without these architectural views it would be difficult to manage the development of any system because these views provide a good framework for managing the design especially considering that we often work with big systems across different disciplines. For that purpose, when working with the system we tried to develop parts of the functional and physical architectural views in parallel once the operational concept was clear.

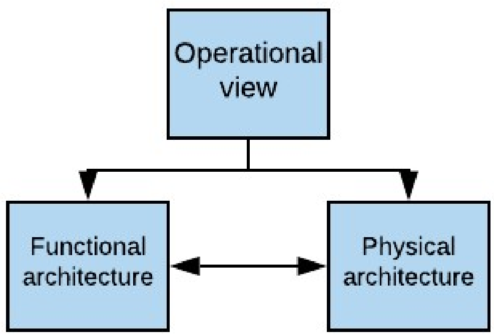


Figure 13: The three architectural views

If we start by looking at the physical view, it's about the physical elements of our system, different components and physical interfaces. It is an important architectural view because here, we elaborate models and views of a physical, concrete solution that answers and to and supports the logical architecture model. Furthermore, the physical architecture will answer to specific stakeholder concerns or other regulations and standards to fulfill the system requirements. A way in which we visualized this architectural view early on in the process was with this block diagram, showcasing the main subsystem/components we wanted in our system.

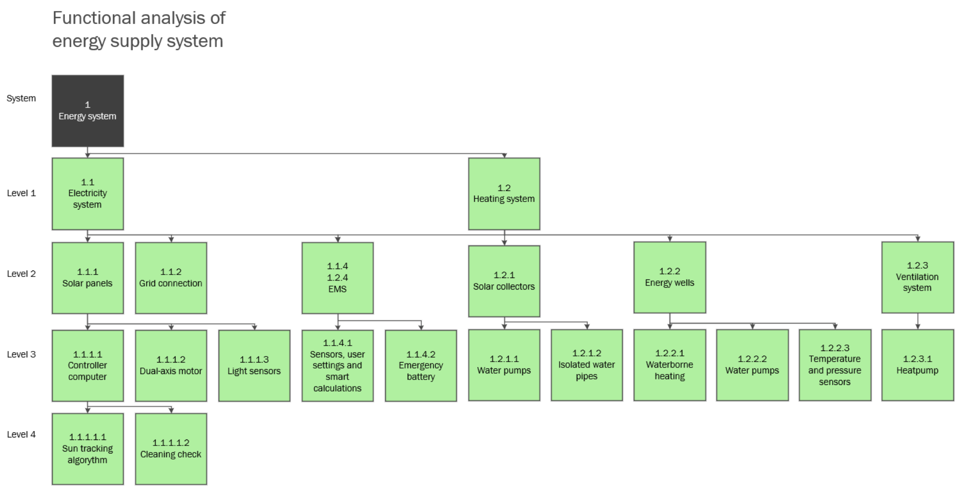


Figure 14: Physical layout

The functional architectural view entails us identifying system functions, their interactions and how the functions will operate together to perform the system mission. It also supports development, along with the physical architecture, of verification tasks that are defined to verify the functional, performance and constraint requirements. With our project we represented this with a functional diagram showcasing the main system functions.

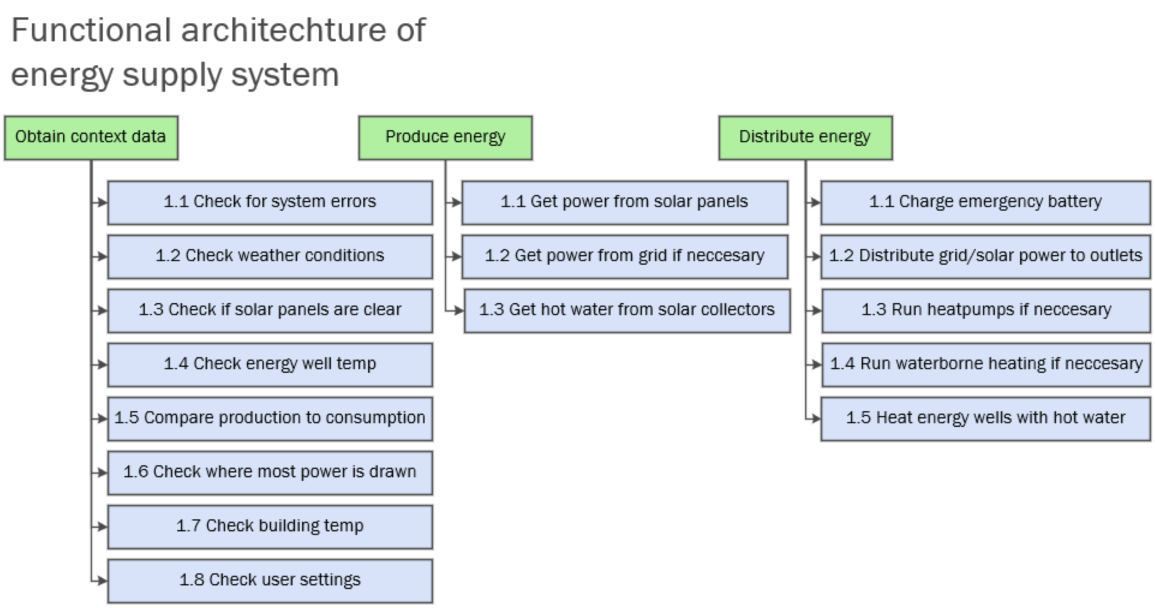


Figure 15: Main Functionality of our system

The functional view also helped us when we wanted to implement new subsystems. By designing use case diagrams, we got a clear vision of how the interaction between an actor and function would go. In this example, we wanted to showcase the different roles of the actors that are interacting with the system. They all have different functionalities they can access, but the janitor for example is limited whilst the GeoStar employee can do a lot.

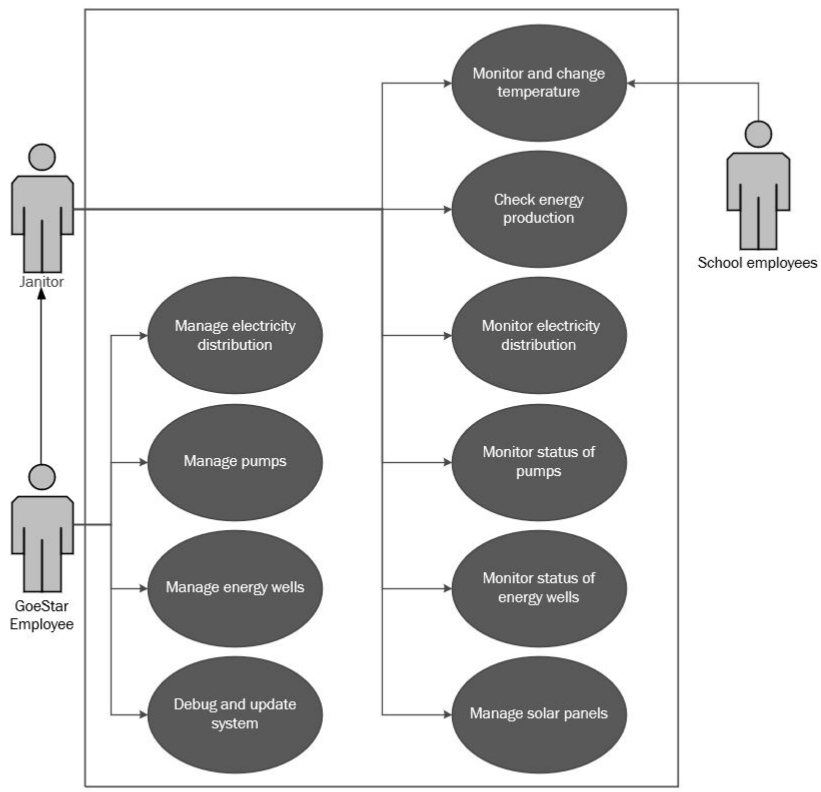


Figure 16: Use case diagram

Looking at the operational view, here we see the system when it's deployed and then how the users will interface with the system. It provides an important understanding in how the design system will operate. Its relationship between the physical and functional architectural view is that it incorporates both views. For us, this view was quite important because it helped us realize how different our system would operate during summertime versus wintertime. To get a better picture of how the system would operate and meet the stakeholder-demands we needed to split the system into two scenarios: summer (May-September) and winter (October-April

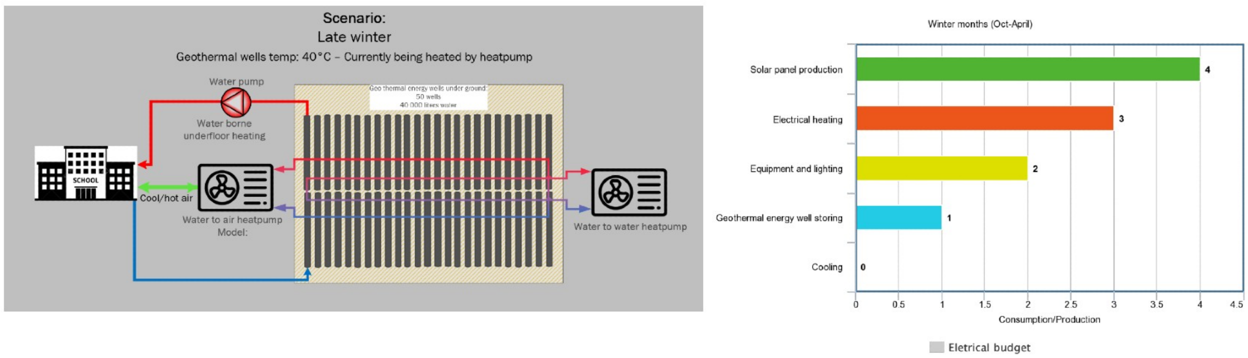


Figure 17: Operational scenario for winter

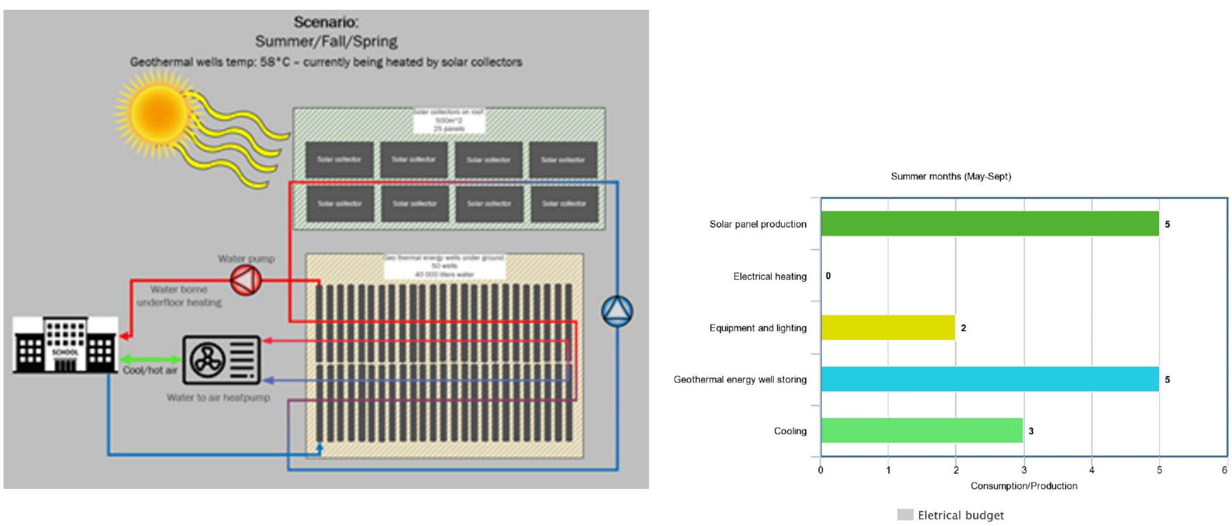


Figure 18: Operational scenario for summer

The architectural view helped us make the distinction between the two operational scenarios by incorporating both the functional and physical view and in turn brought forward different characteristics to our system which we used to understand how our system would adapt throughout the year once deployed.

We also made a 3d model of our system using a program called Solidworks.

It gave us a nice visualization of our system, and it even helped us in the development phase, because in our first iteration of the model, we saw that we could increase the amount of solar panels on the roof in order to "produce" more energy. Having this 3d model therefore helped us both in the physical and operational architectural view.

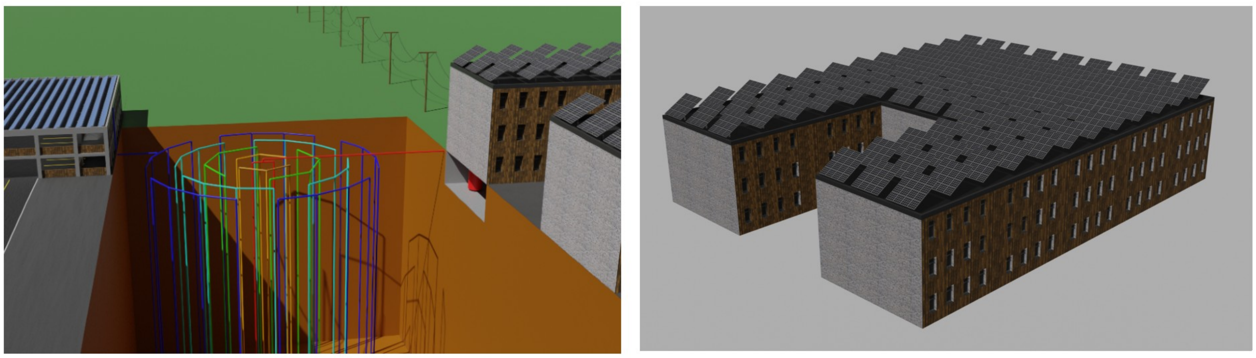


Figure 19: Final 3d model of the system

Chapter 5

Integration, verification and validation

With the system engineering process implemented we are promising our customers and stakeholders several different functionalities, technologies and automated systems. In order to make our promises more than empty words, we have to perform Integration, verification and validation.

System integration is defined as the process of bringing together the component subsystems into one system and ensuring that the subsystems function together as a whole system. It is connected to verification and validation because once the system components have been realized and integrated to form the complete system, we can perform verification and validation, so it's necessary in order to optimize our methods of verification and validation, as well as the method of integration. Validation is often done once the system is installed and deployed but there are many verification activities that we can and should perform in parallel to the integration of the system in order to reduce the number of verification actions and validation actions while controlling the risks that could be generated.

Furthermore, an important distinction between the terms verification and validation must be made because they are often used about one another, but there is a significant difference and the figure below help us illustrate the difference.

Figure 20: Verification and validation

Verification is looked at as the "inner loop", meaning from our implemented design solution, we circle back to our requirements, and ask ourselves the question of whether we satisfied our requirements as they were written. For validation, or the "outer loop", we circle back to our stakeholder analysis and ask ourselves whether we satisfied our customers and stakeholders. We looked at it as, "are we building the right system" vs "are we building the system right".

From the Venn-diagram we see that there are several different verification and validation methods. The methods vary from system to system and they are applicable for different environments.

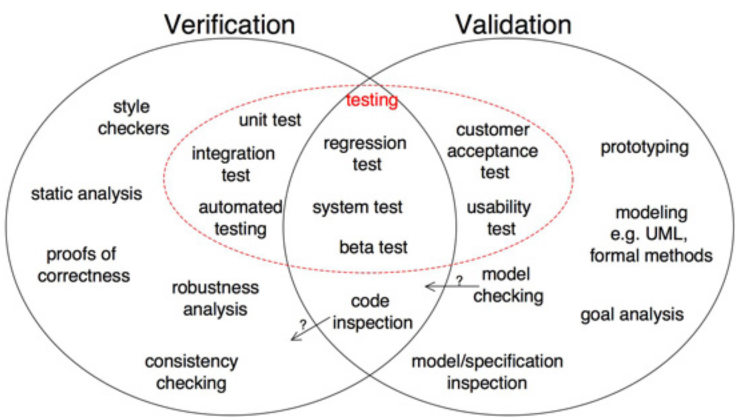


Figure 21: Methods of VV

In our case methods that allowed us to perform verification in the best manner were testing and analysis. Testing allows us to test components and check functionality for subsystems with parameters that are measurable. This is an iterative process which done in parallel with the integration process. For that purpose, we had a test plan that included some selected requirements in the test, as well as the test environment and method to conduct and analyze the test in order to achieve desired results to determine if the test was successful.

The other verification method we benefited from was verification by analysis. Here, the data will be collected and based on this data and our knowledge of the system design, we will make an engineering judgment about whether or not the defined success criteria have been met in order to prove that the verification by analysis was successful. This method is used when we can't do an end-end test or develop a prototype.

In the development, we did actually discuss the idea of creating a physical prototype by using a 3D-printer, but ultimately, we came to the conclusion that for our energy system a prototype would not serve a great purpose. The prototype must show some sort of functionality, and we believed that we would not be able to recreate that in a prototype. If we did create a prototype however, we realized that it would only show the structure of our system, but that could be visualized much better in a 3D model then in a prototype, and for that reason we chose not to produce a prototype and rather perform verification by analysis. Going through these kinds of discussions helped us select the best method of verification for our particular system.

For the validation process of our system, we will do this once the system is installed and deployed because we can't recreate a real-life environment. Once that is established, we will use the following steps:

- Prepare to conduct validation: To prepare for performing product validation, we will use the collected concerns and requirements from our customers and stakeholders.

- Perform validation: For validation testing, we will focus on the expected environment and operation of our system, meaning once our energy system is installed and the school building is complete.

- Analyze validation results: Once the validation activities have been completed, the results are collected, and the data are analyzed to confirm that the end product provided will supply the customer's needed capabilities within the intended environments of use.

- Prepare a validation report: This report provides the evidence of product conformance with the stakeholder expectations that were identified as being validated for the product at this layer. It includes any nonconformance, anomalies, or other corrective actions that were taken.

- Capture the validation work products: These include procedures, required personnel training, certifications, configuration drawings, and other records generated during the validation activities.

5.1 Technical risks

Doing IV&V during the development process also has the benefit of contributing to risk mitigation. In order to be aware of the risks involved internally and externally to our system we developed a risk matrix which starts with identifying the risk, assessing the consequences of it happening, its impact and likelihood and finally developing a risk mitigation plan.

Risk source	Description	Consequence	Impact	Likelihood	Risk mitigation
Technical	System malfunction when starting up	Concern from customers/stakeholders, schedule delay, high cost	High	Medium	Test every componenet individually, then in subsystems and finally as a whole. Frequent technical reviews
Technical	Workers inaccuracy	Can cause wrong handling of the system	Medium	Medium	Provide staff training
Technical	Insufficient energy storage	Can cause the liquid in the wells to freeze resulting in inefficient storage	High	Medium	Have energy available at all times
Technical	Grid power outage	Shortage of power	High	Low	Back up battery
Technical	Battery explosion	Fire, high cost and damage to people	High	Low	Secure battery properly
Technical	Any smart system functionality not working	Shortage of power, reduced efficiency, cost	Medium	Low	Track development, rapid prototyping, proper testing
Technical	Other software system malfunction	Shortage of power, cost	High	Low	Track development, rapid prototyping, proper testing
Technical	Communication interruption/delay	System malfunction, reduced efficiency	Medium	Medium	Test often and early, make sure technology updates are occurring frequently
Technical	Pipe leakage	Damage to property, high cost	High	Medium	Proper insulation
Technical	Limitations in interconnection, grid management, and transmission infrastructure.	Shortage of power, high cost	Medium	Medium	Proper monitoring
Environmental	Flooding	Damage to property, high cost	High	Low	Follow national guidelines, have regular safety training
Environmental	Fire	Damage to property, people and high cost	High	Low	Follow national guidelines, have regular safety training
Environmental	Heavy snow	Damage to solarpanels and solar collectors	Medium	High	Have a cleaning strategy ready for solar panels and collectors
External factors	Stakeholder/customer changing requirements	System changes	High	High	Continuous dialog
External factors	Political instability	Change in system function	High	Low	Up to date with community, country and worldwide situation
External factors	Innovative competitors	Concern from stakeholders/customers	Low	High	Be up to date with smart functionality, research market periodically
External factors	Government restrictions	Can cause operational and functional changes	High	High	Up to date with laws, regulations and standards

Figure 22: Risk matrix

Looking at the technical risks from the matrix, we see that a risk is stated as “workers inaccuracy”. This is a risk because it can cause wrong handling of the system resulting in some sort of malfunction. Unfolding this risk and seeing what requirement it’s associated with, we find the requirement about a “user-friendly interface”.

Having a user-friendly interface is important for the entities interacting with our system, from operators like the school janitor and other personnel, the interaction should not cause any confusion. It’s also important for us as designers because the environment of this requirement affects our whole system, so if we fail at this, then the purpose of our system design is diminished. For that reason, we decided to verify this requirement by designing the interface from the employees’ point of view. As the picture shows it will be easy to navigate with clear colors, text, a consistent menu and more.

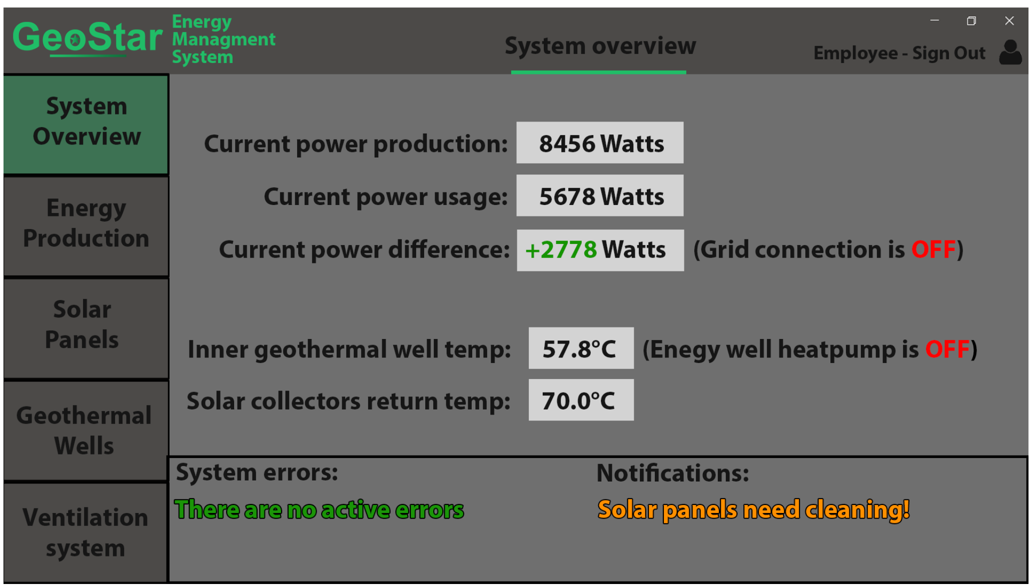
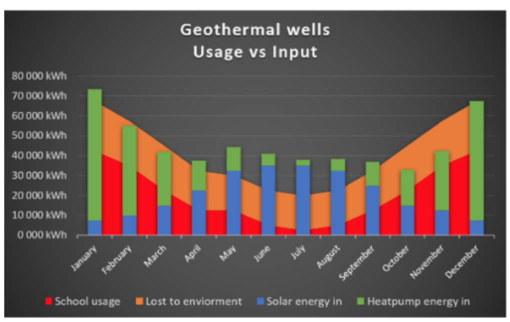


Figure 23: User interface

Regarding the test method for this requirement, we plan to test this by gathering a group of users to operate the system and receive feedback from their interaction that we will use adjust any features for the hope of optimal satisfaction. Validation will be done once the system is installed and deployed in a real environment, so that our customers and stakeholders can try the system and see if it meets their original intent. Before that process, software, different components and sub-systems will go through continuous testing and prototyping in a verification process. This is to ensure that once the system is installed and deployed, that everything goes through as planned. So, the product can’t be validated unless it’s been through our verification process.

Another technical risk is regarding insufficient energy storage in our geothermal wells. Considering this is a large sub-system there are many requirements associated with this technical risk. Selecting two of those requirements, they say that the energy should be available at all times and derived from that requirement is the requirement of having the wells "charged" at all times, meaning energy inputted is sufficient throughout the year.

To verify these two requirements, an end-end test couldn't be performed, therefore we chose verification by analysis and we did this by gathering data on how much energy is inputted to the wells and by which source, versus how much of that energy is lost to the environment and lastly the usage of the school. Most notably, we see from the graph that the input is higher than the usage, which is what we want, especially in the summer times because that's when the solar collectors will experience peak performance and deliver lots of energy into the wells. It verifies that energy will indeed be available at all times. By verifying this subsystem, it will also help us verifying the requirements connected to this subsystem.



Geothermal wells	Solar energy in	Heatpump energy in	School usage	Lost to enviorment
January	7 500 kWh	66 000 kWh	42 500 kWh	25 000 kWh
February	10 000 kWh	45 000 kWh	35 000 kWh	22 500 kWh
March	15 000 kWh	27 000 kWh	22 500 kWh	22 500 kWh
April	22 500 kWh	15 000 kWh	12 500 kWh	20 000 kWh
May	32 500 kWh	12 000 kWh	12 500 kWh	17 500 kWh
June	35 000 kWh	6 000 kWh	5 000 kWh	17 500 kWh
July	35 000 kWh	3 000 kWh	2 500 kWh	17 500 kWh
August	32 500 kWh	6 000 kWh	5 000 kWh	17 500 kWh
September	25 000 kWh	12 000 kWh	12 500 kWh	20 000 kWh
October	15 000 kWh	18 000 kWh	22 500 kWh	22 500 kWh
November	12 500 kWh	30 000 kWh	35 000 kWh	22 500 kWh
December	7 500 kWh	60 000 kWh	42 500 kWh	25 000 kWh

Figure 24: Verifying energy in the wells

By verifying this subsystem, it will also help us verifying the requirements connected to this subsystem. Doing verification this way and gathering data early in the process is important in order to show our customers the meaning behind our words, that the proposed system is based on real data. Not only will it encourage our customers, but us as designers that the requirements and the subsystems we designed were verifiable.

Chapter 6

GeoStar Solutions

6.1 Value proposition

Looking at the value proposition for our system, meaning the benefits our customer has from our system, the main value is that our system is eco-friendly. Using solar energy combined with geothermal wells in a closed loop we have designed an environmentally friendly solution that has zero emissions. We also made a conscious decision to not use any toxic liquids within the wells and the solar collectors that could cause harm to the environment.

Another benefit of our system is that we don't use electricity for heating. Although this can be considered as environmentally friendly, it isn't all good, because heat is considered as a low-tier type of energy because it has less potential for conversion into other types of energy. By implementing a closed loop solution that uses waterborne heating we solved many of these issues and it has proven to be a big benefit from our system.

Another value which the customer will appreciate is the fact that our entire energy system is automated. It contributes to good safety, and by implementing smart technology we will prevent water leaks and damage to important components. The automation results in less physical interaction with the system, but by having a user-friendly interface the customer will still have an overview of the system production and be plugged in to the workings of the system.

Probably the most important aspect for our customer is regarding operation cost. With an automated solution implemented and considering that our main components have a life cycle of 30 years, the system will require minimum maintenance efforts throughout the year and therefore be durable and cheap in operation. This is also shown in the cost of ownership model below.

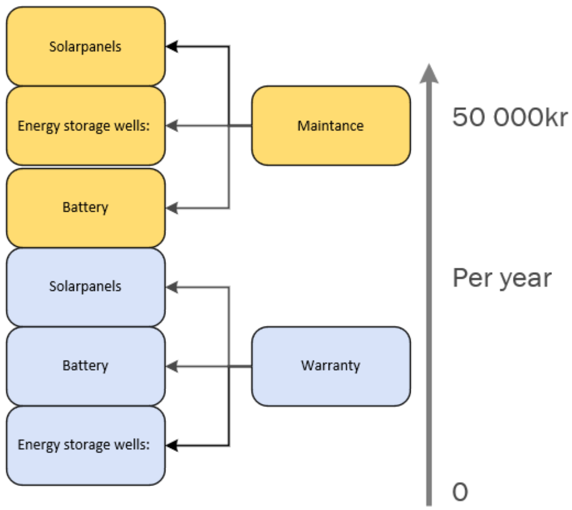


Figure 25: Cost of ownership

6.2 Business proposition

Looking at the business proposition for our company, it is based on designing and selling energy systems for buildings. We work alongside a construction company and oversee the process, but ultimately our responsibility is the design as the business model for our company explains.

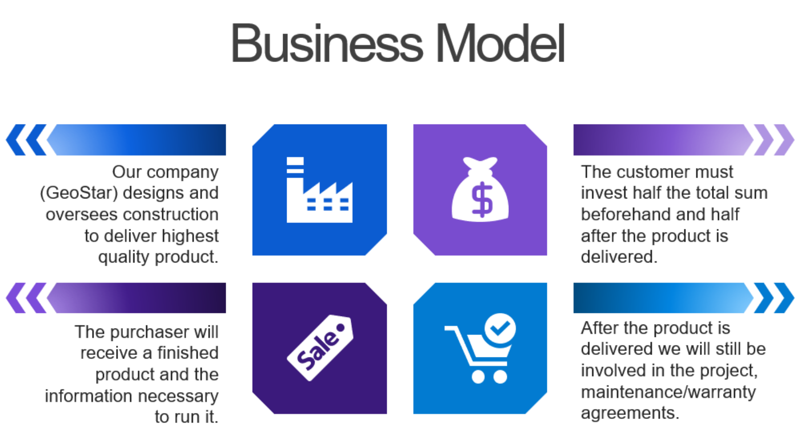


Figure 26: Business model

With the situation in today's world, environmental issues are more important than ever before, pollution is a worldwide problem, and the building industry is a big contributor in increasing the carbon footprint in the world. For this exact reason we recognize a high demand of environmentally friendly solutions, especially in Norway who are a progressive thinking country when it comes to the environment. By proposing our system design to any kinds of buildings, we will contribute in reducing the carbon footprint by using renewable energy.

To be able to deliver an energy system that is competitive in the current market, we need to understand the needs of the market and identify our competitors. In Norway there are several companies that deliver energy systems to buildings. Most of these focus on one of the energy needs a small scale building may have, but leaves out most of the others. This gives us the opportunity to get into the market by creating an energy system that supplies all the energy a building may need. Since most companies target small scale buildings, there are few solutions for bigger buildings like schools, leaving yet another spot open in the market for us to utilize.

6.3 Key performance parameters

Key performance parameters represent those capabilities or characteristics so significant that failure to meet the threshold value of performance can be cause for the concept or system selected to be reevaluated, or reassessed.

KPPs	**Quantified**	**Valid use case circumstances**
Solar energy per year	**300 000-350 000kWh**	Estimated average power output per year in Oslo with solar tracking technology
Solar collector energy per year	**225 000-275 000 kWh**	Estimated average power output per year in Oslo with solar collectors
Energy wells input per year	**550 000 kWh**	Estimated total energy input into the wells per year
Max capacity of wells	**800 000 kWh**	Maximum energy capacity of energy wells
Heat supply to building	**250 000 kWh**	Heat input to school building per year
Building electricity usage per year	**250 000 kWh**	Average power consumtion of the school building a normal year
Liters of water in energy wells	**40 000**	Total liters of water in all the underground boreholes.
Expected life cycle of system	**30 years**	lifetime is expected to be 30 years, yearly service is required.
Weight of solar panels and collectors	**18 tonn/8000kg**	Total weight of solar panel structure and solar collector structure

Figure 27: Key performance parameters

From the figure, a key performance parameter is the solar energy per year is 300 000-350 000 kWh, estimated kWh per year in Oslo with smart solar-tracking technology implemented. This was based on numbers from similar systems in the Oslo area. This is an important parameter considering how central solar energy is to our system.

Another KPP is the total estimated energy input to the wells per year which is at 550 000 kWh. This energy comes from the solar collectors and heat pumps we have implemented. We quickly realized that the efficiency of our system would depend on how the energy in the wells would operate, but with this estimated number and the verification I did in chapter 5 the energy inputted is sufficient.

Lastly, from the figure a key performance parameter is that the expected life cycle of our system is 30 years. As explained, this is one of the value propositions and it is because of our autonomous solution and durable components.

The connection between the key performance parameters and the value propositions is that one can actually see the value and business propositions from the key performance parameters because in fact, they represent the characteristics so significant that failure to meet the threshold value of performance can cause the system to be reevaluated. The KPP about how much solar energy we estimate in a year shows the use of renewable energy and represents a value proposition. For the business aspect, we see that the expected life-cycle of our system is 30 years, showing how durable our company's designs really are.

6.4 Possible challenges

When designing such complex systems, the challenges that might hinder us from achieving value and business proposition can be many. A challenge with the value proposition is that we promise not to use much electricity for heating, but a challenge can occur if there is very little solar energy one year. Obviously, there is not much we can do about that, but it is something to be aware of. Another challenge in the value proposition is that we promise a system which is cheap in operation, however, this can change if there are any internal errors within the systems like communication loss between components or physical damage that might require service and more frequent maintenance.

We also promise our customers a reasonable priced product, but obviously that depends on several factors, such as what components and materials we select for the system. In the design phase we had options of selecting components that had longer life-cycle but because of our propositions to our customers regarding an environmentally friendly solution we had to sacrifice the durability factor for a more eco-friendly solution.

Chapter 7

References

[1] Group 15 (May 2020). Systems Design and Engineering: "Geostar energy solutions" Final report, University of South-Eastern Norway.

[2] BONNEMA, G. Maarten (2017). Systems Design And Engineering: facilitating multidisciplinary development projects, CRC Press.

[3] Monat, Jamie (September 2018). "Applying Systems Thinking to Engineering and Design" https://www.researchgate.net/publication/327758240_Applying_Systems_Thinking_to_Engineering_and_Design

[4] Pinheiro, Francisco (2003). "Requirements Traceability" http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.458.7787&rep=rep1&type=pdf

[5] Inflectra (January 2020). "Requirements Traceability" https://www.inflectra.com/ideas/topic/requirements-traceability.aspx

[6] MITRE (2017). "Interface management" https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/systems-integration/interface-management

[7] Theuni (March 2017). "Internal and External Interfaces From an Operations Perspective (long read)" https://blog.flyingcircus.io/2017/03/30/internal-and-external-interfaces-from-an-operations-perspective-long-read/

[8] Hummell, James (June 2018). "FUNCTIONAL ARCHITECTURE USING SYSML" http://mbse.solutions/functional-architecture-using-sysml/

[9] Faisandier, Alan (May 2020). "Physical Architecture Model Development". https://www.sebokwiki.org/wiki/Physical_Architecture_Model_Development

[10] Faisandier, Alan (May 2020). "System verification" https://www.sebokwiki.org/wiki/System_Verification#Integration.2C_Verification.2C_and_Validation_of_the_System

[11] Steve. (2010 November 29). "The difference between verification and validation". https://www.easterbrook.ca/steve/2010/11/the-difference-between-verificati

[12] Shea, Garrett. (2019 December 16). "Product validation". https://www.nasa.gov/seh/5-4-product-validation [13] Prof. De Weck, Oliver. (2015). "Fundamentals of systems engineering- Verification and Validation" https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-842-fundamentals-of-syste
lecture-notes/MIT16_842F15_Ses9_Ver.pdf

This has surprisingly been one of my favorite subjects, so thanks for that and have a good summer :)