

	Questions	Output of question	RQ1	RQ2	RQ3	Participant category			Additional information about this guide:
Introduction	Explain purpose of this interview, number of questions and approximately time duration Ensure and discuss confidentiality Explain context of software sustainability in our research	Background information				L1-5			Participant categories: L1 = Software developers/engineers L2 = Product owners (?) L3 = IT architects L4 = Infrastructure experts L5 = IT management  Participant categories determine what questions that are being asked
Participant information (open-questions)	Q1.What is your job position?	Background information				L1-5			
	Q2.How long have you worked at the organization? (or within IT?)	Background information				L1-5			
	Q3.What is your job field and experience?	Background information				L1-5			
	Q4. What software applications are you most involved with?	Understand which software the participant is working with. This might affect the answers on how the participant answers on following questions.				L1-3			Research questions: RQ1: Which metrics are best-suited to assess sustainability of a chosen software product? RQ2: What are the sustainability measurements of a chosen software product? RQ3: Can an index established from measurements reveal improved sustainability?
Software sustainability (open and closed-questions)	Q5. What is sustainable software from your point of view? Follow up: a) When can you consider an application sustainable? b) What are the characteristics of software that might indicate that it is sustainable?	Understanding the participant's view on software sustainability and what sustainability dimension is most important. If a participant focus more on a specific dimension of sustainability (e.g technical), their answers on questions related to metrics and measurement might be focused on that direction	X			L1-5			Have you think about the providers of hardware and how to choose based on sustainability?
	Q6. How important is software sustainability in your working context? (1-5 scale, where 5 is very important)	Understanding the participant's awareness and experience with software sustainability. Might affect the participant's answer on other questions related to metrics and measurement tools	X	X		L1-5			
	Q7. From a strategic perspective, how can you work and improve software sustainability?	Understand how software sustainability is being prioritized from a managers perspective. Might affect the participant's answer on other questions related to metrics and measurement tools				L4-5			
Metrics and measurements (open and closed-questions)	Q8. Is sustainable software addressed in your projects/portfolios in a standardized and systematic way? Follow up: a) If yes, can you explain how you address it? b) If no, how would you address it in a standardized and systematic way?	Understanding how participant wants or already develop, use or assess/measure sustainable software	X	X		L1-3			
	Q9. Where in the software development process should you think about sustainability? (from manufacturing hardware to usage of the software) (Alt. If you consider energy consumption, where would you think about this in the software development process?) Follow up: a) How would you improve software sustainability in the process?	Understanding how software sustainability can be improved and where in the software process it is easiest to do it. Can link their suggestions to metrics discussed in Q10-14	X			L1-3			
	Q10. Have you or your team been measuring or assessing sustainability of software? Follow up (if yes): a) What type of software did you measure? b) What metrics/KPIs did you use and were the metrics general, domain-specific or application-specific? c) How did you choose metrics? d) What tool did you use? e) What was the result from the measurements?	Understanding how software sustainability can be measured and what metrics and tools that can be used. Also, understanding if generic metrics were used or adapted to the measured software product's characteristics	X	X		L1-3			
	Follow up (if no): f) How would you measure or assess sustainability of software?								
	Q11. Which direct and indirect metrics do you think are important to measure sustainability of an application? Follow up (if any metrics are given) a) What do you need to measure the sustainability of software? b) Can these metrics be used to any type of software?	Understanding what metrics that can be used to measure software sustainability, if they are generic or application-specific and what tools that can be used	X	X		L1 & L4			
	Q11-How feasible are the following metrics to measure software sustainability? (1-5 scale, where 5 means means very feasible)?		X			L1 & L4			
	Q12-How actionable are the following metrics to measure software sustainability? (1-5 scale, where 5 means very actionable)?		X			L1 & L4			
	Q13-How accessible are the following metrics to measure software sustainability? (1-5 scale, where 5 means very accessible)?		X			L1 & L4			
Closing questions	Q14. Before we end this interview, is there something you would like to add related to metrics and measurements of sustainable software that we have not talked about yet?					L1-4			

	Participant: Mathijs	
Question:	Answer:	Highlighted:
Q1	IT engineer. Officially I am an IT engineer, that is my function since last year June. Work with data engineering, software engineer and the other position is that I unofficially have is that I work with the sustainability dashboard. For that I do a bit of solution, data and software engineering.	IT engineering
Q2	9 months	
Q3	3 teams: Data engineer, Software engineer, Sustainability dashboard	
Q4	For the data engineering part, I work with the International Data Platform applications mostly. At the moment, I work for the Java side on International Web Application to implement new requirement and also how to migrate that data from on-premise data center to the Cloud.	
Q5	How green it is. Minimise energy consumption, without unnecessary tasks. How maintainable software is, leads to more sustainability. Lot of interest in green solutions. How better you are at software developing, the more efficient you want you code. It's not the main focus for a lot of people. They look more at optimising the performance, regarding the sustainability. <b>Relative sustainability: look at percentage of utilisation of hardware.</b> Absolute sustainability: <b>how much energy the software consumes</b> , not easy factor to compare applications	
Q6	4 - important, very busy with it	
Q7		
Q8	Sustainability is not standardised within workflow, more for cost reduction. To achieve this: make insightful, set goals. For relative sustainability it is already in place but more for <b>cost reduction</b> .	Only standardized for cost reduction practices
Q9	<p>Sustainable software from my point of view...with a definition of looking at how green it is (because you can look at how maintainable it is) it is about minimizing energy consumption as well as not havng extra, unnessecary performance. I have also heard other people talking about "Sustainable software" as how maintainable it is. True sustainability is mainly looking at the power consumption. I try to write <b>efficient code</b> in an optimal manner which has a direct correlation to the energy being consumed. I am interested in sustainability and some developers just focus on building things that are working. One part within software engineering is to do things more efficient, then you start thinking you want things to run as fast as possible and then you also start look at price optimizations. The better they become, energy consumption is decreased. Main focus of <b>other people might be to create faster application and they start over optimization. For example, if someone creates a website and users want to access that website. Maybe they do not look at user's requirement of page load, which then leads to they opimize a page load to 0.5 seconds when 1 seconds is enough from the user's perspective.</b> And that over optimization is something that happens alot. Because they only focus on creating good applications, but they do not take account what the actual requirements are to create applications with a good energy consumption ratio.</p> <p>There are 2 seperate things, relative sustainability and absolut sustainability. My definition of sustainability is to look at the absolut part at power consumption. For the absolut, <b>how much energy is my hardware consuming because of my application.</b> You can take other things into account as well, <b>how efficient the hardware is that you are running things on.</b> As a developer, you dont have that much control of this part. The absolut is just measuring a certain number, for example <b>Watts of energy is being consumed when running the application. The larger the application, the larger amount of energy is being consumed.</b> It is not an easy factor to compare different applications but it is a good metric still. The relative part, <b>percentages of how well the hardware is being used.</b> Looking into the rightsizing of the hardware, try to use the hardware as effectively as possible. Dont have actually energy consumption numbers.</p> <p>Standardized way for <b>cost reduction part, which can be related to software sustainability.</b> Would address it in a standardized way by transparancy. They are <b>now measuring the absolute part</b> (power consumption) in the bank. The relative part is important (utalization and that kind of things) but they are mainly looking at it from a cost perspective.</p>	Important with <b>software architecture</b> . E.g. choose a server that is only being used when the application is needed.
Q10	Relative sustainability is being looked at already. For making absolute sustainability insightful, work is being done at this moment. Clean up is very important to avoid resources being used for nothing. Monitor applications is also very important, to see if new tactics work. General metrics are used: <b>Utilisation - CPU, Storage; Energy consumption - kWh; Carbon - co2 per tons.</b> Those metrics are available, and important and required to use for development. Not looking at how efficient the code is, no standardised way for within the bank. If the pillars are combined, <b>code efficiency</b> can be determined and applications can be compared. SonarQube for code optimisation, let senior developer look at the code. The dashboard with absolute values gave insights in checking if applied tactics are more sustainable.	How much power is our application using. Clean up resources that is not being used. Used <b>general metrics</b> that were accessible (things like utilization) when measuring the POC, not application specific. <b>Measured energy consumption in kilowatt per hour.</b> Utilization metrics can be used to <b>measure sustainability of the hardware</b> , but not for how the code is running. We have no standardized ways of measure the code. <b>Hardware utilization + code optimization + absolut energy consumption values</b> = measure sustainability of an application
Q11		

Transcript	<p>Sustainable software from my point of view...with a definition of looking at how green it is (because you can look at how maintainable it is) it is about minimizing energy consumption as well as not havnig extra, unnessecary performance. I have also heard other people talking about "Sustainable software" as how maintainable it is. True sustainability is mainly looking at the power consumption. I try to write <b>efficient code</b> in an optimal manner which has a direct correlation to the energy being consumed. I am interested in sustainability and some developers just focus on building things that are working. One part within software engineering is to do things more efficient, then you start thinking you want things to run as fast as possible and then you also start look at price optimizations. The better they become, energy consumption is decreased. Main focus of <b>other people might be to create faster application and they start over optimization. For example, if someone creates a website and users want to access that website. Maybe they do not look at user's requirement of page load, which then leads to they opmize a page load to 0.5 seconds when 1 seconds is enough from the user's perspective.</b> And that over optimization is something that happens alot. Because they only focus on creating good applications, but they do not take account what the actual requirements are to create applications with a good energy consumption ratio.</p> <p>There are 2 seperate things, relative sustainability and absolut sustainability. My definition of sustainability is to look at the absolut part at power consumption. For the absolut, <b>how much energy is my hardware consuming because of my application.</b> You can take other things into account as well, <b>how efficient the hardware is that you are running things on.</b> As a developer, you dont have that much control of this part. The absolut is just measuring a certain number, for example <b>Watts of energy is being consumed when running the application. The larger the application, the larger amount of energy is being consumed.</b> It is not an easy factor to compare different applications but it is a good metric still. The relative part, <b>percentages of how well the hardware is being used.</b> Looking into the rightsizing of the hardware, try to use the hardware as effectively as possible. Dont have actually energy consumption numbers.</p> <p>Standardized way for <b>cost reduction part, which can be related to software sustainability.</b> Would address it in a standardized way by transparency. They are <b>now measuring the absolute part</b> (power consumption) in the bank. The relative part is important (utalization and that kind of things) but they are mainly looking at it from a cost perspective.</p>	
------------	---	--

	Participant: Arthur	
Question:	Answer:	Highlighted:
Q1	IT engineer, data engineering. Programming and developing software, works mostly with cloud software	
Q2	1 year	
Q3	Involved in two fields, software and data engineering	
Q4	Since working on development a lot: VScode	
Q5	Software that does not consumes a lot of <b>energy</b> , <b>efficient coding</b> , sustainable <b>how data is transfered between server and client</b> , size of data. looks at it from the enviromental perspective. Electriscity usage.	Efficient coding, energy consumption
Q6	4, since I am interested in sustainability and the enviroment. Although, it is not being addressed in the team as much. I did not learn anything about IT and sustainability when I started at the bank it's something that came across along the way.	
Q7	Had a green coding project where I learned more about the mindset related to software sustainability, not very specific.	
Q8	Sustainability is not standardised in day-to-day work, not directly. Would be good is it is. Coaching could help, educate personnel on the topic. SonarQube for checking code on the sustainability is an option.	<b>Measure how long it takes for a user to make a task or a specific workload (related to usability)</b>
Q9	At the design of the application. Change in mindset on the development side.	Imrove sustainability in the development stage or the workload for the client
Q10	Do not have the tools to measure software within the teams, this is worked on. (I would assess it how I am developing it)	
Q11	<b>How fast the code runs, cpu-usage/utilization, how long it takes to perform a task.</b> These metrics come to my mind while I am coding myself, some of these metrics are generic but for specific application you might need product-specific metrics	
Transcript	<p>My job at ABN is mostly working on programming and development of software, working through all lifecycles projects and application. So from the start of development till the deployment and maintance of the projects and application and I work mostly with cloud applications. Software sustainability for my point of view is software that does not consume too much resources such as electricity. Also software that is <b>efficient coded including the data transfers between server and client</b>. For example <b>large data transfers consume more energy</b>. I look at it from an environmental perspective. Things that indicate that an software product is sustainable might be the <b>electricity usage and the maybe total energy used to render an front page (front-end)</b>, this may be hard to measure. <b>Time to interact, if an user wants to create an account the amount of clicks consumes energy, more clicks means more resources consumed</b>. Also the <b>page weight the lower the better</b>. The importance of sustainability within my context is 4 because I am really concerned about the environment and how we produces a lot of CO2 and damaging our environment. Although the IT-sector is not the number one sector that produces carbon footprint it can help reducing the carbon production. The sustainability topic was something I learned throughout my job, when I joined it was in its starting phase and learned a lot throughout the green coding project at the bank. qSoftware sustainability is actually not really described in an standarized way in our job we focus more on the desires of the client we should pay more attention to the sustainabililty and come up with an standarized way. We could do this from the beginning and start teaching the colleagues how to work on a more sustainable way. For the coding we have SonarQube which helps with coding in a more sustainable way we should widely use it since its basically uploading code and it checks automatically. At the development and the client usage we can improve on lowering the <b>energy consumption</b>. In the current projects in involved with we are not doing any measuring because we also do not really have the tools. Mathijs and the team of Kasper are actually working on the measuring tools to measure the sustainability but me personally do not measure. I would assess it on how I am developing the project, so <b>if the code is taking a lot of CPU too run</b> or maybe the <b>data transfers</b>, I would like to measure this. It would be able to implement calculators to see how users use the application for example you could <b>keep track of the clicks used for the tasks</b>. An metric could be how <b>fast the code runs</b> or <b>how much CPU</b> it uses or <b>how long it takes for a user to perform a task</b>, these are things that pop up in my mind during developing code. These things might be <b>applicable for all types of software but there will also be specific things you could do for some applications depending on the type of application</b>.</p>	

	<b>Participant: Bas</b>	
<b>Question:</b>	<b>Answer:</b>	<b>Highlighted:</b>
Q1	IT-Lead for the international full banking and clustered data	
Q2	15+ years with ABN not only within IT	
Q3		
Q4	International core banking team, cluster stage base.	
Q5	<b>efficiently coded programs using languages which are efficient in nature</b> (energy consumption), how its coded and how its used. If performance is not key on the agenda, it is easily overlooked at. Sustainable software is also <b>reusability</b> (find reusable components) which also contributes to low maintenance cost, saving development and testing time. Also infrastructure it runs on. Less coding to do, less people that needs to do it. We have people working in India that goes to work on not a very sustainable way. So <b>when less developers is needed, it also can save CO2 emissions</b> . Underlying infrastructure is not doing that much. Performance that is seen by the users is sometimes scaled to high, if user requirements are being considered. In IBM we cannot scale, but in Azure it is better manageable.	Reusability, efficient coding, resources when developing the software, performance, user requirements vs scaled up performance
Q6	4 - Important, likes the topic a lot. It can not always be the highest point on the agenda. Sometimes the priority lies somewhere else. - We have goals, dashboards, management attention.	
Q7	You need to measure it, everything and quickly because only then you can see if you achieve your objectives -> get insights and come up with how to work with it. If you want to outsource your IT, in the end of the day it is your IT so if you are you using a polluting IT vendor, it will also effect your IT even if it is outsourced. You need to balance it within the IT-landscape.	
Q8		
Q9	When working on a solution think about both the cost and co2 impact. When buying software, reviews in how sustainable the software is, is required on forehand. When building software it should be in the design fase. Also see if it can be validated during the testing fase. During the design phase of the infrastructure and the applications. Measure CPU, memory, underlying infrastructure, resources for processing time, Amount of data	

Q10	<ul style="list-style-type: none"> <li>- Code and how it is used</li> <li>- Performance -</li> <li>- How is the software is being usage</li> <li>- Reusability (API, circularity)</li> <li>- Doing a task takes too long (look at the underlying infrastructure)</li> <li>- Cannot scale within IBM or in their own data centers</li> <li>- Direct metrics + indirect metrics ()</li> <li>- Procurement (buying software)</li> <li>- CPU, memory</li> <li>- Processing time for CMS</li> <li>- Employees doing things (building software, go to the work etc.)</li> </ul>	
Q11	Underlying infrastructure metrics, CPU utilisation. Process lead time, volumes in data in gb's or mb's. Work with patterns, are basically building blocks. This is useful for reusability.	

	Participant: Wiebren	
Question:	Answer:	Highlighted:
Q1	IT architect	
Q2	22 years in IT, 12 years in ABN.	
Q3	No expertise to be honest in a specific technical field. Connecting people and teams within the bank. Story teller, connecting dots. Insights from both business and IT. Also now on the social side and the regulatory side	Story teller, connecting dots
Q4	Not a software developer. Involved most in infrastructure decisions, both for on-prem and off-prem. Hardware infrastructure or virtualizations infrastructure. Real applications I am not so involved with anymore. 2015 I was more involved in markets applications. At the moment it is more infrastructure.	
Q5	I think it will be software that is here and has a <b>dynamic</b> footprint, comes and goes with the demand if that makes sense. And also software that <b>can be u sed for a long period of time</b> , not written for maybe half a year. But software that runs for years or maybe decades that delivers a certain function. <b>Composability</b> and <b>maintainability</b> of software is also a big part of the characterisitcs that enables it to have a long lifetime. <b>Scalability, dynamic resource allocation, writing in a language with a lot of knowledge.</b> Certain languages are more or less used. If you write in Cobol for the mainframe, it is been a lot of people retiring and it is difficult to get people want to work in Cobol. So if you talk about maintainability or adjusting the software for something, it can have a longtime, knowledge needs to be there. Do not go for very nished languages (if it is not for a sustainable purpose). It is about circularity on a software level. <b>Reusability. If code is really good written and creates a certain functionality or component, why not reuse it.</b> Maybe a green software section in this kind of Github repositories. Software needs to be <b>delivered in a modern way, be able to run in different platforms and enviroment - compatibility.</b>	Dynamic resource allocation, reusability, composability, comptability, scalability, and maintainability.
Q6	5 - very important, main drivers in work at this moment.	
Q7	At this moment not, more driven by other factors than this challenge (sustainability).	
Q8	Start in different phases and in different roles. Requirement phase - how green should the solution be. Starts with awareness and critieras questions. Design phase - Apply most green designs and solutions, keep it simple. Operations phase - make sure you keep on delivering on those. In all phases, changes should be made of the software development.	Focus on sustainability in all phases

Q9	<p>Think about software sustainability in the <b>design phase. I think it is the most important phase.</b> If you don't integrate it in the design phase, it will be difficult to achieve it in the operational phase. I would focus on <b>scalability</b> and <b>elasticity</b>. I think a lot of resources for both software and hardware (and computing) caters for that at the moment. Scaleable compute with containers. On-demand compute serverless. It is all technology that can drive down footprint. <b>Limit data growth.</b> Challenge because only more and more data is being generated. One data generates this data and you analyze the data and you backup the data etc. Exploding world of data. Data needs to be stored and as soon as it exists, it has a footprint. And until you delete it, it keeps having this footprint. You can maybe lowering it a bit by moving it etc. Data itself is the new oil, not only for the good reasons but also for the bad reasons because it has a footprint. For me <b>data is actually something to be very careful with for different reasons, also for social reasons and privacy reasons.</b> But also data as a footprint. How can we make sure that IT and data is used in the benefit for the human kind. This is a big challenge for IT to address and that is why I often say that IT is can be a solution to the problem but it is also a part of the problem. This is a challenge for IT. Maybe you want the data for security reasons, but how big is the change that you will actually use it. Trade-off at the moment (e.g. redundant data in banks for security vs footprint). If you are not addressing real society problems, what are you really doing. <b>Maybe a company problem sounds interesting, but the true judgement should be made if you are contributing to society or not.</b> A company is only a virtual construct. Access the flow and the storing of data.</p>	<p>Easiest to make a difference in the design phase -&gt; scalability and elasticity. Limit data, only generate and use what is needed. Address real society problems. Less is more!</p>
Q10	<p>Some initial steps. Eco-chain report, on a high level. Very general, all levels are required. Workspace devices, back-end applications, data centers, cloud, SaaS. Done some first steps to create transparent level. Now we want to go in a deeper level to see on a workload level to give insights to people who develop or are in charge of these operations/functionalities/business logic. Metrics used were very general and absolute we need to go on a business, grid and application level. Also maybe transaction level to give each transaction a footprint. We have calculated kWh per Co2 within Azure or power consumption in the data centers. We also want to measure on workload level. We choose <b>Co2 and circularity levels</b> because we have two targets on those. On Co2, we came to the conclusion that we need energy consumption. For circularity, it is focused on material usage and if it can be reused, also from IT assets.</p>	<p>Metrics used were general. kWh per Co2 within Azure or power consumption in the data centers. Focus on Co2 and circularity.</p>
Q11	<p>What metrics are valid or important (set of measurements), can we identify them and relate them/weigh them for different conditions or companies? It is a part of the sustainable quality model that we want to develop. I would now say biodiversity situations. <b>Environmental side: material usage and energy usage.</b> Hardware device, I seen hardware run for 50 years. I have systems up and running for 10-15 years. Even the last couple of years they would be the most stable ever. Trade off between circularity and energy. Material running for a long time but the energy usage might be a bit higher. There is a rapid replacement of hardware going on, 3-4 years and the devices are gone again. Back-end is more about 6-8 years (longer). Massive amount of computer storage, massive amount of materials going into this hardware -&gt; replacements are enormous which gives a footprint. <b>Circularity levels, energy consumption. Material usage and energy usage.</b> I have also seen applications or hardware where I can't touch because people do not know what runs on it. Things keep on running without people know if business needs it. Is it still needed? Rational your IT landscape, activate what you really need and be critical about it. <b>Less is more!</b></p>	<p>Material usage and energy usage.</p>
Extra		



<p>If a data center uses green energy, can a software application be considered sustainable?</p>	<p>There are more factors to consider. It gives kind of, indeed a distorted image I think. There is actually a second picture in the Ecochain report that is also relevant - CO2 related to the Grid mix. The data centers scores so low is because they purchasing 100 percent green energy. Therefore the CO2 image is very low, but they still use energy (that comes from the Grid ofc), so the Grid mix is therefore what actually happens on the power supply side. If we compare public cloud (70 percent renewable energy) with on-premise (100 percent), you get a distorted view indeed in how they relate to each other. If we would apply off-setting, public side could go down further than it is now. What is the reality there? We need to normalize it by using the grid mix on the country side. The <b>grid mix</b> in Netherlands is this... if you purchase 100% this or if you do off-setting, it can get a distorted view.</p>	<p>Many factors to consider except for just how green energy is in the data centers. CO2 related to Grid mix.</p>
--	---	---

	Participant: Rajiv	
Question:	Answer:	Highlighted:
Q1	Solution engineer	
Q2	Since 2004	
Q3	Has been primarily in the banking domain	
Q4	Azure platform, using java and T24	
Q5	It seems there are various aspects: <b>development lifecycle, how much energy that is software is consuming when users are using, how efficient is it, acceptance within the industry, user experience. Technical dimension: production usage. usability, production lifecycle. The amount of effort creating software</b> should also be considered in sustainability of software. <b>Development and production</b> should both be taken into account. <b>Programming language</b> is an aspect. Should be a balance in the main focus of a company. Java is primary language. Small patterns to <b>reuse</b> within the organisation. Reduces the development time, more sustainable. <b>How much effort did it take to build the software, e.g 1000 developers working spending too much time on code or product. Production lifecycle it is sustainable but in the development, its not very sustaniable.</b> Development perspective – <b>Some code language that are more efficienct. Usability</b> can leverage sustainability. <b>How much time to we spend on building the software.</b> If you deploy a software at it runs on green energy, is it sustainable or not?	Reusability, code language and user requirements, usability.
Q6	3/4 - important. Slowly getting into the system.	
Q7	First thing to do is know how much you're spending. This is currently being developed.	
Q8	It has to drill down from the top. But also, the <b>choice of language for development</b> based on research, which language are more optimized. More awareness for the <b>reusable components</b> . Boils down to how <b>efficient coding</b> you can do. Difficult to address.	
Q9	<b>Language choice, reusable components</b> you use, <b>get a good grip of requirements - design phase.</b> Leads to an efficient development. If you understand the requirement, you build the software and the business will not accept it. IChoice of algorithm, takes time to get the mindset to think about this within the organisation. Identify the smallest set of integers. Pick up an algorithm which is more optimized rather than choising the one that results in the same conclusion. The more optimized application, the more sustainable. Cannot influence hardware (strategic decions to use Azure). The only area we <b>could understanding requirements, code language and reusable compentant.</b>	
Q10	Not on day to day level.	
Q11	<b>User friendly</b> helps also towards sustainability. <b>Time a user spends on a task. How many man-hours have been spend during development.</b> Less bugs. How efficient is your test coverege ( <b>less bugs</b> )/ How good the software is tested. <b>Acceptance – meeting functionality or not.</b> Has to be developed again which in turn requires energy.	Number of bugs, task execution time, man-hours for development, user requirements
Transcript	In a java driven development, if we have to build an application that meets one of the business requirements, one way is to build everything on your own. Instead, you can reuse patterns, components, into primary applications.	

Ramon		
Question:	Answer:	Highlighted:
Q1	I am in a department that used to be called the center of expertise of software development. Our job is to help departments with their code quality. Recently I started coding communities for all the developers within ABN. These community contain around 3000 people who have bi-weekly meetings. These meetings are about different topic regarding code quality which also contains sustainable. I personally do not measure any software but I know that department are working on this topic.	has a community that share information around sustainability and other development topics, Automated tool for code quality such as SolarQube
Q2		
Q3	Community Manager	manages communities within the company to improve code quality
Q4	More involved with programming languages such as Python, C++ and Java	
Q5	From the developers POV an sustainable software is use less energy for the same results. One scientific measure and the best one we got is the <b>KWh</b> a software consumes. You have have a specific workload and some test scripts. Test and see if one application uses half of the kwh, it can be considered twice as efficient. Although its difficult to compare two applications. We have an infinite things to do with an application, it is not like a car. <b>Run application on a server, and than you process a workload, that is energy consumption in kwh.</b> For me <b>sustainability is only focusing on the energy.</b>	
Q6	Sustainability is like 3 percent of the things I do. I do think its extremely important though. I am very interested in measurements.	
Q7	Less code you write is better. Have a critical view on frameworks usage and avoid it as much as possible. Framework requires more storage, memory. It takes time and it adds complexity for the developer and more work for the machine.	
Q8		
Q9	<b>Less code is more.</b> Using frameworks might improve the sustainability of the code. <b>Reducing the line of code.</b> The whole scope is important, we need to find where the most energy is consumed and reduce that. If you would have an overview of all application and how much energy it uses, you can remove those that dont use a lot. Which applications use a lot? We need to discover where the energy is used to try reduce it.	Should consider the whole scope, you should find where it's the consumed most and we should fight it.
Q10	Getting the data from the datacenter on the applications, the racks and from Azure. I think this is the first step, then we have to do some iterations to see what's interesting there. To measure sustainability software: Best thing is kWh, difficult to compare. Mostly focusing on energy.	
Q11	<b>Reduce code</b> , more than 1 millions lines of code. Metrics, measure all applications. Focus on applications that consume the most energy. Auto scaling in cloud, If application has multiple instances it also needs to be scaled down again. If a resource is idle you need to pay but is not using much power. <b>Code quality and complexity can be measured with sonarqube, the amount of dependences</b> , if an applications has too much, the applications can be very inefficient. Mocking in code can be an indicator. <b>Compare energy of application with revenue of application</b> , combine financial part. Utilization of resources. How many that are booked can be an indirect metric. For power consumption, if this resources are idle. You are paying for it. It is less efficient powerwise. But regarding sustainability, it is still sustainable since it is not using much power. Code quality. The amount of lines of code in an application combined with the functionaly support - Cyclomatic complexity. Amount of libraries that an application has. If an application has 300 dependencies, what kind of applicaiton is that? Complexity metric. Causal relationship between inefficient application and application that has bad quality metrics. <b>How many mocks in code is an indicator of how efficient the code is.</b> Performance wise it makes it build very slow.	Idle time: code quality; amount of dependencies; amount of mocks, energy consumption, utilization of resources, amount of lines; amount of libraries

Transcript	<p>Sonarcube has certain rules. If you select everything from a database, maybe that is not what you want. Maybe you just want a small piece. We get offers from different vendors that we evaluate. There are a lot of initiatives within sustainability. We started a month ago or something. We are in an analysis phase. We start getting measurements but we don't have it yet. Leon has a POC first with some measurements. We have not done any measurements yet. Imagine you have multiple applications, how are you going to define how efficient an application is. It is really difficult to measure. With an electrical engine, it is easy to measure the energy. If you look at the useful output, and what you put in it. Then you get a percentage that is converted to useful work. But if you have an application for mortgages for example. It starts using less power because it has little input.</p>	
------------	---	--

Participant: Dorothee		
Question:	Answer:	Highlighted:
Q1	CIO of Private Banking, Risk and finance domain of IT	
Q2	1.5 years at ABN, Sustainability was the reason I came to ABN	
Q3	Chief Information and Digital Officer	
Q4	<p>Sustainability is more than reducing CO2 but also different topics of ESGs are involved. Three different domains are involved:</p> <ul style="list-style-type: none"> <li>• <b>Social</b></li> <li>• <b>Environmental</b></li> <li>• <b>Governance</b></li> </ul> <p>We focus more on the Environmental part. This dimensions also consists of three different domains:</p> <ul style="list-style-type: none"> <li>• <b>Personal</b> <ul style="list-style-type: none"> <li>o Our personal responsibility in regards to how do we protect the environment with our behavior. You can think of how do travel (fly), how do we eat, how do we act. We should really start acting instead of still talking because we are losing generations.</li> </ul> </li> <li>• <b>CEO Responsibility</b> <ul style="list-style-type: none"> <li>o I am responsible for how IT is done. Transform the IT organization to protect the environment. Technology and Digitalization is a strong lever to change to whole company. We should look how to reduce the overall <b>power consumption</b> with partners within the whole ecosystem.</li> <li>o The IT stack from the application to the hardware, we need to optimize through the layers.</li> </ul> </li> </ul> <p><b>We should flag or categorize the application with CIA. I would like to know how environmental friendly an application is so it needs to be measurable.</b> How much infrastructure resources and CO2 does it produce. Rating the application as a <b>Quality Flag. (red, yellow, green)</b> --&gt; here comes your study into play. There are a lot of <b>coding practices</b> we should work on and implement. (Querying example given, use more specific query in order to get the data faster). <b>We are good on measuring the infrastructure, this includes hardware, storage and VMs an application consumes.</b></p> <ul style="list-style-type: none"> <li>• What do we develop as solutions to our customers (end or business customers) for their transition to sustainability. <ul style="list-style-type: none"> <li>o What is the application enabling? What is the positive impact of the application? Environmentally friendly, produce what is needed nothing more. <b>Green by software.</b> Look at the IT impact and what is being achieved with it. Not only, how do I run it, lifetime, how often do I need to update it. End-end sustainability case.</li> </ul> </li> </ul>	<p>Teams should be coded so it uses minimal energy. By using Teams we get people from planes and journeys. Should not only restrict it to an application, <b>how much can we reduce energy consumption</b> VIA the application in the whole ecosystem.</p>

Q5	<p>Reducing CO2 Emission. ESG. ABN is more focused on the environment side also myself. Three dimensions, 1. its our personal responsibility to protect the environment by our behaviour, Creating awareness in your environment 2. CIO is responsible how IT is done. another responsibility is protect the environment as much as possible. 3. what do we give our customers to develop more sustainable. App that calculates ESG and selects companies that are sustainable? Try to improve <b>infrastructure solutions</b>. On-prem or off-prem?</p> <p>Hardware level, circularity</p> <p>How is the application designed? <b>Architecture level</b>. Coding practices. Optimize the coding.</p>	<p>don't believe in a subgoals, do not constrain yourself just to IT. We should look for the overall picture, not just the energy consumption of an application but what is also reduced, like less flights (teams). We need to optimize applications through the whole stack. Applications are rated based on the CIA triad used to see which are most important ones. for ABN is that apps with high C I A. P-rating -&gt; privacy rating. ABN tries to optimize the infrastructure resources consumed. We need to use and develop coding practices to make applications build and run better.</p>
Q6	This is not asked but we can make a safe assumption its very important so 5	
Q7	We should change our behavior and should look into what kind of impact IT can have on sustainability	
Q8	?	
Q9	One should work on all three dimensions. Social, Environmental and Governance	
Q10	<b>Simpler the landscape is</b> , the less cost – maybe more sustainable? Drag down complexity, <b>simplify</b> . We carry too much legacy that includes that from 10 years ago that we still need to maintain. This leads into running old applications, so we should simplify which reduces the cost.	
Q11		

Participant: Sam Price		
Question:	Answer:	Highlighted:
Q1	Product Owner of the PCSnextGen platform	main reason between PCS and PCSnextGen is moving everything to azure but it needs a lot of refactoring not just moving it.
Q2	4 months at ABN Amro	
Q3	Infrastructure Architect	
Q4	All the workloads move to azure in the long term. PCS support cloud native functions, to make the migration easy. Currently there is no application running on the PCS nextGen at the moment.	
Q5		
Q6		
Q7	Monitor <b>power usage</b> , check suppliers. Lower the cost perspective, also higher sustainability.	
Q8		
Q9		
Q10	No measuring done yet from the datacenter perspective.	
Q11	Energy the most easy one. Split up on platform themselves, use tags. Relative straight forward. For hardware it is harder, depends on what they offer. Application level should be straight forward, from a power level. Vm dedicated to application. Pcs next gen will be owned by ABN, cms is owned by kyndryl. Next to energy consumption, make the whole process visible using blockchain, very transparent and accurate.	
	no involvement in measuring	
	plan to do it in datacenter mgmt team	
	I would use the <b>energy from the PDU</b> , whole lifecycle of the hardware.	
	PCSnextgen will be owned by kyndryl and VM - they own the platform and we just use it.	
Question:	Answer:	Highlighted:
Q1	Design of pcs next gen platform.	
Q2		
Q3		

Q4	All move to azure in long term. Pcs support cloud native functions, to make the migration easy. 0 applications on pcs next gen at this moment.	
Q5		
Q6		
Q7	<b>Monitor power usage</b> , check suppliers. Lower the cost perspective, also higher sustainability.	
Q8		
Q9		
Q10	No measuring done yet. From datacenter perspective.	
Q11	<b>Energy</b> the most easy one. Split up on platform themselves, use tags. Relative straight forward. For hardware it is harder, depends on what they offer. Application level should be straight forward, from a power level. Vm dedicated to application. Pcs next gen will be owned by ABN, cms is owned by kyndryl. Next to energy consumption, make the whole process visible using blockchain, very transparent and accurate.	



Participant: F		
Question:	Answer:	Highlighted:
Q1	Value Cluster Lead of International Core Banking	delivering all systems to international, every system outside the netherlands is delivered by me
Q2	since the beginning 2018	the functioning was called something else but they are doing it now for 4 years, was not always in IT, when in and out then starting with retail, when to operations then back to IT
Q3		
Q4		
Q5	Sustainable software is for the lifetime is easily expanded and the extensibility is well spend since its usable for a long time. You should scale it to the right proportions.	we should be careful with lazy programming, this could poor code quality. object oriented programming referring to the same class and reusing the same classes. using a lot functions which is also unmentainability.
Q6	2 - Money and Time is more important	
Q7	look at the footprint during the development of software, during debuggy grading on the efficiency and effectiveness of the software (idk if this is possible)	create awareness around how much we are using software, storage etc. Do we really need them, frequently checking. Emphasizing that we are going for the efficient solutions not simple solutions.
Q8	on the computing side we are, we have the energy consumption its a parallel connection, but I am not looking at the electrical bill. Resource Cost vs Electrical/Sustainability Cost there is a big difference. this is 1/20 of the whole cost.	Velocity of what the teams should be producing, administration of how much the team predicted and compare it to other teams
Q9		
Q10	Efficiency and Productivity is measured. Sustainability and maintainability yes. CO2 consumption we only do for a part in the cloud. All applications in bas domain.	how many people do you need, how many infrastructures you need.
Q11	We measure piece of software how much it goes through the pipeline, looking to check on the vulnerability in the delivery itself. We moved the all test environments to the saas subscription. i don't know if it very sustainable.	
Transcript		Sustainable software should be easily maintained. By doing so, you don't need huge investments for the future. You only need one-time investments and its well spend because you can maintain it for a longer period of time. Also scale it to proportions that is needs - scalability.