Methodology Essay Summary Examples

METHODOLOGY SUMMARY

This is a summary of the methodology essay

TypeScript is a superset of JavaScript developed and maintained by Microsoft. This means it has the added benefits of TypeScript but the ability to transcompile into JavaScript (Bright, 2012). TypeScript's most notable feature is its addition of optional static typing to the language and static type checking.

The industry recommends having a "strict configuration" in the TypeScript options file. This forces you to utilise all the features of TypeScript and ensuring every element has a determined type. It also recommends not using the "any" type. A type that states at any variable can have any value. Using TypeScript and the "any" type is pointless, and you might as well just use JavaScript if you are going to ignore TypeScript's benefits.

TypeScript also offers a lot of benefits not available in JavaScript. These include enumerated types and class interfaces. Powerful features that make TypeScript important for enterprise software development.

During the project, I utilised TypeScript and found that the most notable feedback I got from fellow developers was the need for declaring types for everything. This ensured that I knew the "shape", or interface, of every element I was interacting with.

I also took advantage of enumerated types to ensure I didn't have magic values throughout my application. A good practice for enterprise-level developers to ensure values aren't magically appearing throughout an application's code.

Ultimately, it was determined, through guidance from other developers at Jade Software and following the guides created by the TypeScript creators, that I was following TypeScript's best practices and taking advantage of TypeScript many benefits over JavaScript. TypeScript will be a language I will utilise well into the future because of these benefits and the increased need for it in the industry.

SUMMARY OF METHODOLOGY ESSAY

The methodology chose for the project was ITIL's Change Management process which I utilised to review the ICT Division's practices.

Given the large scale of ITIL's Change Management process, I concentrated my essay on seven aspects that ITIL suggest should be part of every request for a change to the IT environment. These elements help to ensure that every change is assessed for their risk of impacting the IT services and business units. Change management can benefit an IT department in numerous ways including fewer incidents caused by changes, a higher success rate of changes and fewer calls to the Service Desk about negative impacts caused by changes that were implemented. The business ultimately benefits by having lower service disruption, more communication about change, and continuous improvement to their services.

Analysing Ara ICT against this framework highlighted where some good practices have fallen through the cracks. Though risk management is performed, there is a misunderstanding that change management is about the diagnosis of errors that are caused by changes rather than managing change such that these do not occur.

However, staff are committed to keeping the IT environment secure and stable. They recognise the benefits of good change management and so have been ensuring that risks are managed, although only informally. Their communication with business is generally good, and they always take measures to ensure that changes do not have an unexpected impact.

For the Ara ICT Division to improve and grow their services to the business, they will need to shift the focus to the idea that change management is used to prevent damage to the live environment, rather than as a cure.

Methodology Report

The methodology analysed for this project was The Nine Boxes, an agile micro interviewing technique. The Nine Boxes allows the interviewer to collaborate with the interviewee to drill into issues being experienced, identify who is affected, and ideate solutions. This technique is made of 9 boxes that an interviewer will iterate throughout during an interview.

	What is the Problem?	Who is Affected?	Visualize the future.
OPEN	1	4	7
CONTROL	2	5	8
CONFIRM	3	6	9

Figure 1 Example Of The Nine Boxes Interviewing Technique

There are several benefits to utilizing this technique. It provides a framework to guide and direct interviewees, and ensure the required topics get touched on throughout a discussion. For a new interviewer, it was useful to be able to refer to a table of questions if there was a lull in the conversation. The emphasis on ideating a solution and querying both open and closed questions was also useful during the project. This technique is definitely useful for starting requirements and allowing the interviewer to capture multiple dimensions of an issue.

The Nine Boxes does have some limitations – it only provides some options on how to start questions but does not implicitly reference interviewing best practises. This is what makes it a micro technique, as it is not useful as a stand-alone interviewing technique. Additionally, the framework can be difficult to follow as the natural flow of most conversations veer across multiple topics at once and therefore struggle to organically stick to a framework.

Overall, The Nine Boxes is a handy micro technique that has much to offer in terms of managing and directing interviews but is not robust enough to stand alone. Having a framework to refer to is useful, as well as varying between open and closed questions; But best interviewing practises are also required. I would use this framework again, but in a looser capacity.

METHODOLOGY ESSAY SUMMARY

The methodology I explored during the project was best practices for a RESTful API, a set of design constraints used when creating web-based API's. My research into this topic helped influence my decisions during the prototyping phase of the project as it helped convince me that the best approach to the problem would be to use a web-based user interface. And although the theory suggests that all six of the REST architectural principles must be followed for a system to be considered truly RESTful, there is no reason why it cannot be successful as a partial REST system. Out of the six principles the API I created for my project only follows three, this does not make it a terrible implementation though by any means. The other three principles did not apply to the project in any meaningful way and shouldn't be implemented just for the sake of fulfilling an architecture. The RESTful design constraints outline how a large-scale API should be built to optimise efficiency and scalability but that doesn't mean that some of the principles can't be applied in a useful way to smaller projects. The use of a client-server architecture has allowed for an easily distributable front-end, essential for fulfilling the projects ease of access requirements. Following a consistent interface enabled me to create an effective framework to build project upon. And by using a stateless system the server side of the solution is less complex and easier to build in a short amount of time.

Summary of Methodology

Applications of Software Design Patterns in the TechLabs Project

Design patterns are proven solutions to resolve the design problems frequently encountered by the developers. Each design pattern consists of four elements: Intent, Motivation, Structure and Consequences. Generally, the patterns are categorised into 3 groups, creational, structural and behavioural; these groups represent the types of design problems the pattens try to solve. Design patterns can help developers identify the objects in the system, decide their contents and their relationships with other objects. Many patterns use interfaces to resolve design problems while providing the flexibility to making change. They also increase the efficiency of communication and documentation. However, design patterns are not well-received by everyone. One of the reasons is ill-implemented patterns could deteriorate the codes instead of improving it.

In the TechLabs project, there are 3 patterns applied to the developed applications. These patterns include strategy pattern, builder pattern and factory method. Strategy pattern was used to provide different logics to create scripts for different operating systems. Builder pattern was utilised to create complicated VirtualMachine objects instead of using the class constructor. Factory method was applied to separate the creation of the concrete strategy objects from the other classes.

Overall, using the patterns did resolve the design issues I encountered and increase the efficiency of development. However, each pattern also brings new side effects and they are not necessarily positive to the existed system. Thus, when using design patterns, developers need to consider the contexts and adjust the implementation according to maximise the benefits of the patterns.

SUMMARY OF METHODOLOGY CONCLUSION

Due to an individual project with a one-person team project, the implementation of the IEEE 802.1X project did not align with the Tait Agile-based method. I had to take on both the Product Owner and Development Team roles and responsibilities. I needed to perform to ensure the results were aligned with the initial project requirements and made my own decision on identifying the best way to achieve the goal and kept on top of my schedule, my workflows and my deadlines.

From my point of view, the idea of combining the roles of the Product Owner and Development Team into one person would only work for an individual project. This was because the tasks involved were less and easy to be managed. In a group project, I would hesitate to recommend this practice as it would reduce the Scrum principles of focus and accountability.

Overall, it can conclude that project management using Agile Scrum methodology has revealed many benefits. Particularly with the Scrum process, the involvement of the cross-functional team and a series of sprints that make the information transparent to the Development Team members, so that they can make quick changes and adapt to customer needs.

However, the implementation of the IEEE 802.1X project did not quite follow the Scrum framework but the IEEE 802.1X project had adapted the Agile Scrum methodology and realized many benefits of Scrum.