Chapter 4:

1. **Are IT applications an asset or expense?**
   IT applications are an asset. They are system created to support the business in the functions it needs to do. For example, the ability to originate a loan, to input request information, to process a loan decision, etc. Businesses, like IVK, capitalize on software making it another form of an asset.
2. **What is the main purpose of allocating IT costs to user departments?**
   The reason IT costs are allocated to user departments is that we can see how much overhead is being used for the department on financial statements. It helps to calculate profitability which can be used to further fund new activities.
3. **What is an appropriate percentage of the IT budget to spend on maintenance?**
   There's no right answer to this, but ideally, you'd want to spend as little as possible on maintaining and as much as you can on providing new services. Though this is harder to achieve on larger projects.
4. **As a percentage of sale, ho much should a company spend on IT?**

   About 5% of sales should be spent on IT. A higher growth rate however sometimes calls for more investment in IT.

Chapter 5:

1. **How do IT investments create value or enable value creation?**
   IT investments allow for firms to consolidate market share and assume a more dominant position in an industry. It helps companies to get an edge over their competitors. Especially IT creations that can't be copied by competitors.
2. **How might we get a quantitative handle on the level of value provided by IT?**
   One way to get an idea of the value IT provides is through asking business units, the ones that are customer faced, what allows them to win deals or what helps them become more profitable or gain market share. Getting direct feedback from business units will show what role IT play in supporting their needs in order to do their jobs properly. Therefore, IT ROI can be calculated by seeing what revenue business units bring in as they rely on IT to close the deals.
3. **What light does the consultants' report shed on the matter of Davies firing and the subsequent choice of Barton as CIO? What (if anything) does it add to the IT value discussion?**

   The report showed that Davies was struggling with the transitioning to Type 2 manager involving managing portfolios of projects and infrastructure and being a senior team leader. He also had difficulty interacting with peers and the people he reported to. Davies enjoyed staying in his comfort zone of IT when faced with problems on the business side of things. Because of these insufficiencies, the CEO found that Barton was better suited at the business/management side of things and then took a different approach to the position of CIO. Instead of a tech savvy person to be CIO, he chose a business savvy person to be CIO instead, filling the holes the Davies couldn't. This shows that IT cannot be valued just by looking at itself. IT itself does not directly generate revenue as it is not customer facing, but when looked from a larger point of view it can

be seen that it provides the support to get a large ROI. Similarly, to get the best value out of IT, IVK didn't need someone who just knew the world of IT, but someone on how to manage it and make it more useful.

Chapter 6:

1. **Which side would you take in the debate between Huerta and Calder?**
   I would take Calder's side as I agree with the points she brought up. With my experience in the workforce, I find that requirements are always changing and the customer's needs are as well. I agree with the point that not everything can be anticipated and so using this method, problems can be solved during the devolvement process as they arise. Additionally, by creating prototypes the customer can get a better sense of the end product, how it will look, how to use it, etc. This can allow the customer to bring up any potential problems at an earlier stage rather than at the end when a final product is made. I find that sometimes planning discussion take up so much time only to be changed later that they weren't worth the time and resources.
2. **Does it make sense to jump directly into project coding and early prototypes in order to discover "messes" (unanticipated issues) early – that is, "to fail fast to succeed sooner"? Can this advice be implemented in a practical way?**
   It does make sense because problems always arise during projects. However, this way they will appear sooner rather than later when perhaps the project is almost completed and it would take a lot more work to make a change. Additionally, messing up a prototype would not be as costly or time consuming rather than messing up a final version of the product. This advice can be implemented using the agile work method where tasks are done incrementally using stories and changes are expected. Changes can then be added more easily.
3. **How can you manage, or prepare to manage, what you cannot anticipate and do not expect? What do you think of the approach that Davies seems to have used (as reveled in the documents discovered by Barton) to manage uncertainty in IT projects?**
   You can use the agile work method to prepare for uncertainties. This iterative method allows for early warnings on problems and different opportunities you can take to work around problems that come up. Davies used a system where he'd mark up the estimate of time needed to complete a project based on the confidence level of that project. A high confidence project needed a lower markup while a low confidence project needed a higher markup. This way, Davies accommodated for any unexpected problems that would arise. I think this somewhat works but I would not use it to manage uncertainties because a less intense deadline leads to less priority or slack on the project. If there is no sense or urgency then the work produced will reflect that. Most of these projects almost always took all the time estimated with the markup.

Chapter 7:

1. **What root causes led to the need of a major infrastructure replacement project?**
   Reasons for an IR change include outdated middle and back-end systems. For a company to grow and expand, they need up to date systems running the company. If they stay out of date on their technology, they will lose their competitive edge on their competitors and they will lose market share.

2. **What makes such projects so difficult?**
   A large project like this has many parts and challenges. If you change one thing you have to change everything connected to it. Breaking the project down into smaller chucks is also really difficult as you'd be replacing an infrastructure that's an immense tangle. You can't cut one part and just work on that; it is connected to multiple things.
3. **Do you agree with Barton's decision to terminate the agreement with NetiFects? Can projects like the IR project be avoided?**
   I agree with Barton's decision to terminate the agreement with NetiFects. They're spending too much time and money on switching/adapting to technologies IVK have that the actual IR work isn't getting done. It's a good choice especially if there are other companies out there who offer the same service with the technology IVK works with. It seems NetiFects also have a side goal of converting IVK to Unix based systems and pressuring them through inefficient work on their part with the IR project. Projects like IR can't be avoided since updating middle and back-end systems is vital for a company to continue to grow and expand. But it does not need to comp at the expense of 3 million dollars when there are multiple options available in the market.
4. **What are the advantages of all-at-once versus piece-by-piece projects? Which would you prefer?**
   All-at-once projects can be more useful for continuous development without interference as the requirements and limitations are already outlined and now the project just needs to be made. It saves time in the actual development phase. Piece-by-piece projects work iteratively so that changes can be accommodated for more easily. They have a shorter planning phase and a longer development phase. I prefer the piece-by-piece project as changes happen often and requirements change so a piece-by-piece project is especially useful when the customer is still unsure of the final product they desire. If a change needed to be made to an all-at-once project, it would be a bigger problem for example if it was structured like the IR project. If there were multiple connected parts all of them would have to change.