**Antworten zu den Fragen, beantwortet durch Simon aufgrund der Kapitel 2 und 3 :**

* a)  What were the main reasons the project failed?
* b)  Why is software complex?
* c)  Why is software engineering challenging?
* d)  What are the implications for software engineering?

**Antworten zu den Fragen, beantwortet durch Dominique aufgrund der Kapitel 4 und 5 :**

* *a)  What were the main reasons the project failed?*

Es wurden Teilbereiche aus dem Leistungsumfang gestrichen, dies deutet darauf hin dass der ursprünglich ausgeschriebene Leistungskatalog zu umfangreich und zu komplex war. Zusätzlich wurden mehrfach die Methodik als auch die Software gewechselt und insgesamt gesehen gab es in den Schlüsselpositionen zu viele personelle Wechsel, erfahrungsgemäss führt jede Neubesetzung zu weitreichenden Änderungen.

Von Beginn weg wurde es verpasst eine Bedarfs- und Ressourcenplanung zu erstellen, als wichtiges fehlendes Teil im Puzzle der Softwareentwicklung kann hier auch das Fehlen von Meilensteinen erwähnt werden. Durch alle diese Umstände entstand eine viel zu grosse Projektorganisation die zu viele Entscheidungsträger enthielt und zu wenig ausgelastet war. Dadurch entstand zeitweise eine Führungslosigkeit innerhalb des Projektes und es entstand eine Eigendynamik bei den externen Mitarbeitern.

Weiter kommt hinzu dass diverse Vorschriften umgangen wurden und Aufträge nicht korrekt vergeben wurden. Begründet wurde dies mit dem Termindruck, eine korrekte Ausschreibung der Aufträge hätte aber vielleicht dazu geführt dass durch Rückmeldungen bereits früher klar geworden wäre dass das Projekt nicht umsetzbar ist in der vorgesehenen Zeit.

* *b)  Why is software complex?*
* Sobald die Anforderungen steigen wird es schwierig eine Software einfach zu gestalten. Vielfach werden die Anforderungen mit Individualentwicklungen befriedigt. Aber auch aus einfachen Bausteinen lassen sich beliebig komplexe Strukturen aufbauen. Aber zuerst braucht man einfache Bausteine.
* Ein weiteres Problem komplexer Software entsteht bei der Entwicklung. Oftmals vertragen sich unterschiedliche Werkzeuge, auf die Softwareentwickler zurückgreifen müssen, untereinander nicht. Daraus entstehen endlose Programmiercodes und ausufernde Softwarearchitekturen welche zu einem sehr hohen Risiko für Fehler führen. Diese Fehler sind in komplexen Applikationen meist nur von Hand aufspürbar, was aber leider meist nicht zu 100% glückt.
* Das Hauptproblem komplexer Software sind letztlich die ausufernden Kosten.
* *c)  Why is software engineering challenging?*

Zum einen aufgrund der Anforderungen durch den Kunden. Wenn, wie im Fall von INSIEME, zu viele Anforderungen gestellt werden ist eine Zielerreichung fast unmöglich. Besonders schwierig wird es wenn die Kundenanforderungen nicht mit bestehenden Modulen abgebildet werden können sondern Individualentwicklungen notwendig werden.

Hinzu kommt das nicht immer einfache Zusammenspiel diverser externen Berater und Lieferanten. Passendes Sprichwort dazu wäre wohl „zu viele Köche verderben den Brei“, insbesondere wenn diese aus verschiedenen Küchen stammen.

Weiter ist es wichtig von Beginn weg, also während der Initialisierung eines solchen Projektes, eine saubere und vollständige Ressourcenplanung zu erstellen. Nur so kann sichergestellt werden dass zu jeder Zeit des Projektes genügend Mittel zur Verfügung stehen.

Auch ein Technologie- und Methodikwechsel innerhalb eines laufenden Projektes birgt sehr hohe Risiken. Jeder Wechsel fordert viel Zeit und das Projekt muss zwangsläufig in Verzögerung geraten.

* *d)  What are the implications for software engineering?*

Eine erste Implikation der Softwareentwicklung ist sicherlich im Hinblick auf die alternativen Entwurfsoptionen zu finden. Ein Problem oder eine Anforderung kann auf unterschiedlichste Art und Weise gelöst werden. Den „richtigen“ Weg zu finden ist schwierig, zumal der Entwickler meist nicht die Endusersicht hat. Zusätzlich sollte eine Lösung auch die Implikationen der Alternativen im Hinblick auf Kohäsion und Kopplung berücksichtigen. Die beste Alternative sollte also auch eine gute Kohäsion und eine hohe Stabilität haben sowie skalierbar sein.

Die mit Abstand grösste Implikation geht aber zweifelsohne vom Kunden aus. Der zunehmende Wettbewerbsdruck zwingt die Unternehmen ihre Geschäftsprozesse immer schneller und auch häufiger anzupassen. Dies hat eine direkte Implikation auf die Softwareentwicklung. Der Kunde erwartet eine konstant hohe Qualität, aber auch die notwendige Geschwindigkeit um mit seinen Prozessanpassungen schrittzuhalten.