CS466 – SOFTWARE PROCESS

AGILE & ITERATIVE DEVELOPMENT (CHAPTERS 5&4)

A MANAGER'S GUIDE BY: CRAIG LARMAN

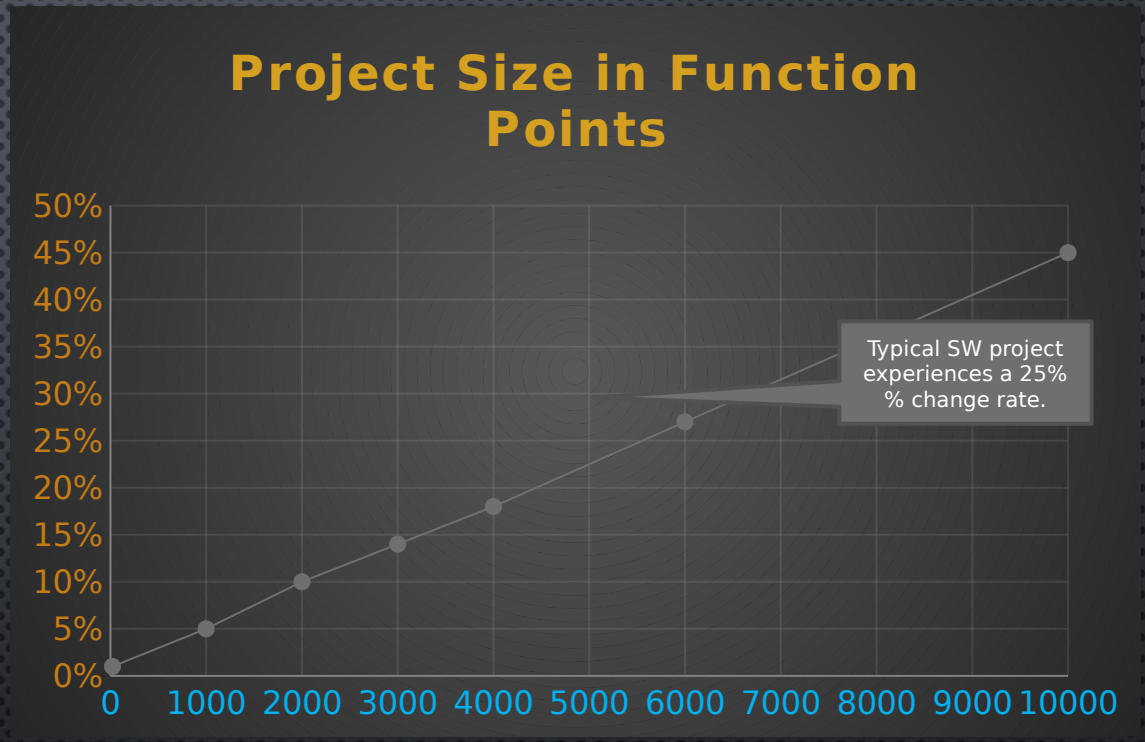
WEEK 5: MOTIVATION & AN AGILE CASE STUDY

BY: JOSEPH MARTINAZZI

THE FACTS OF CHANGE ON A SW PROJECT

THE FOLLOWING GRAPH IS BASED ON RESULTS FROM MULTIPLE LARGE-SCALE SOFTWARE DEVELOPMENT PROJECTS. [JONES97]

- IT ILLUSTRATES THAT AS THE COMPLEXITY OF THE PROJECT INCREASES (FUNCTION POINTS) THE AMOUNT OF REQUIREMENT CHANGE (OR CREEP) ALSO INCREASES.
- MEDIUM SIZE PROJECTS HAVE A CHANGE RATE OF 25%
- LARGE SIZE PROJECTS HAVE A CHANGE RATE OF 35%



THIS ENFORCES THE CONCEPT THAT AN ITERATIVE LIFECYCLE MODEL HAS A BETTER CHANCE OF SUCCESS THAN A SEQUENTIAL LIFECYCLE MODEL SINCE IT CAN ADAPT BETTER TO CHANGING REQUIREMENTS, FOCUSES ON ARCHITECTURE AND HIGH RISK REQUIREMENTS EARLY, HAS A BETTER PRODUCTIVITY RATE, AND A PRODUCES A HIGHER QUALITY PRODUCT (ONE WITH FEWER DEFECTS).

KEY MOTIVATIONS FOR ITERATIVE DEVELOPMENT

Iterative life-cycle models compared to sequential life-cycle models	
The iterative life-cycle model is lower risk	*compared to the waterfall life-cycle model.*
The iterative life-cycle model is designed for early risk mitigation and discovery	*compared to the waterfall life-cycle model.*
The iterative life-cycle model supports the high-change nature of software development	*compared to the waterfall life-cycle model.*
The iterative life-cycle model builds team and customer confidence as production code is incrementally released	*compared to the waterfall life-cycle model.*
The iterative life-cycle model provides opportunity to demo the system to other potential customers	*compared to the waterfall life-cycle model.*
The iterative life-cycle model provides more relevant project tracking	*compared to the waterfall life-cycle model.*
The iterative life-cycle model provides a higher quality product with less defects	*compared to the waterfall life-cycle model.*
The iterative life-cycle model provides a higher probability that the final product will be want the customer wants	*compared to the waterfall life-cycle model.*
The iterative life-cycle model better supports the concept of continual process improvement	*compared to the waterfall life-cycle model.*
The iterative life-cycle model requires more customer engagement, resulting a better probability of success	*compared to the waterfall life-cycle model.*

KEY MOTIVATION FOR TIMEBOXING

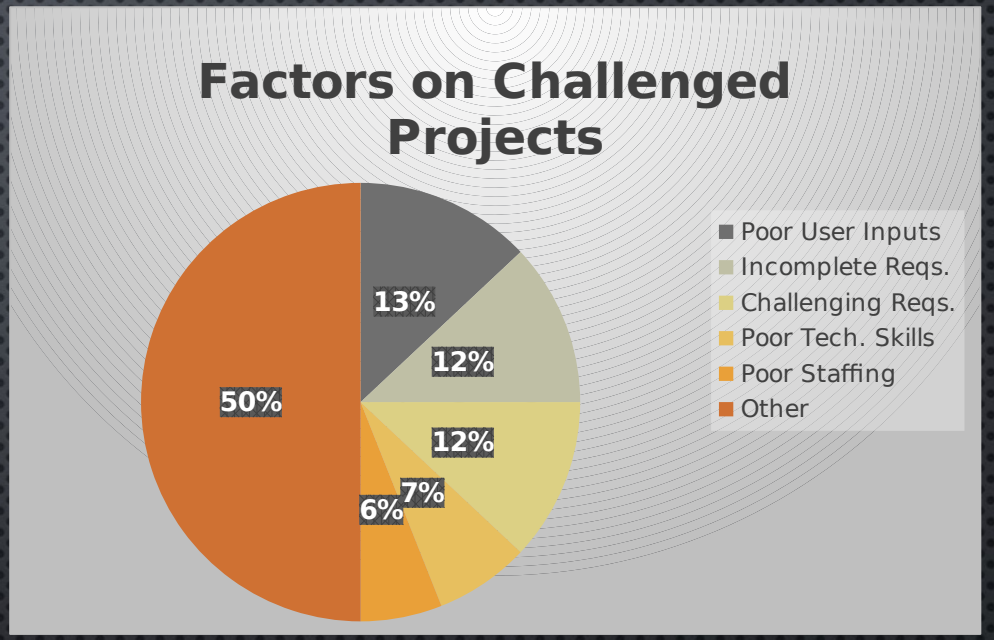
- THE PRACTICE OF TIMEBOXING INCREASES PRODUCTIVITY AS A RESULT OF FOCUSING THE TEAM ON THE END DATE OF THE TIMEBOX. THE AUTHOR STATES THAT TIMEBOXING MAY BE VIEWED AS AN ANTIDOTE TO PARKINSON'S LAW: "WORK EXPANDS SO AS TO FILL THE TIME AVAILABLE FOR ITS COMPLETION." [PARKINSON58]
- ANOTHER BENEFIT OF TIMEBOXING ITERATIONS AS WELL AS THE ENTIRE PROJECT IS BECAUSE PEOPLE REMEMBER SLIPPED DATES, *BUT NOT SLIPPED FEATURES*. EVERYONE WILL VIEW A PROJECT THAT SLIPS 3 MONTHS HAVING 100% OF THE FUNCTIONAL AS A "FAILURE", HOWEVER THE PERCEPTION OF A PROJECT THAT DELIVERS 75% OF THE FUNCTIONAL ON TIME MAY BE CONSIDERED A SUCCESS *IN SOME CASES (E.G., WITH CUSTOMER BUY-IN)*
- ANOTHER BENEFIT OF TIMEBOXING IS IT FOCUSES THE TEAM ON TACKLING SMALL LEVELS OF COMPLEXITY WITHIN A SHORT PERIOD OF TIME.
- ANOTHER BENEFIT OF TIMEBOXING IS IT ENABLES EARLY FORCING OF DIFFICULT DECISIONS AND TRADE-OFFS.

MEETING THE REQUIREMENTS CHALLENGE ITERATIVELY

IN A STUDY OF OVER 8,000 SOFTWARE PROJECTS, 37% OF THE FACTORS ON CHALLENGED PROGRAMS RELATED TO REQUIREMENTS AS SHOWN IN THE GRAPH ON THE RIGHT (POOR USER INPUTS, INCOMPLETE REQUIREMENTS, CHALLENGING REQUIREMENTS). [STANDISH94]

IN A STUDY OF FAILURE FACTORS OF OVER 1,000 SOFTWARE PROJECTS, 82% OF THE PROJECTS SITED REQUIREMENTS AS THE NUMBER 1 PROBLEM. [THOMAS01]

VARIOUS OTHER STUDIES SUPPORT THE FACT THAT REQUIREMENT CREEP IS A LARGE CONTRIBUTOR TO PROJECT FAILURE.



PROPONENTS OF THE WATERFALL METHOD – TYPICALLY POINT TO THIS REASON AS WHY IT IS ESSENTIAL TO FREEZE REQUIREMENT DEVELOPMENT UP-FRONT.

HOWEVER, THIS IS EXACTLY WHY ITERATIVE INCREMENTAL DEVELOPMENT OF REQUIREMENTS WORK – IT FORCES THE CHANGE TO OCCUR EARLY IN THE PROJECT, THUS MINIMIZING THEIR IMPACT!

PROBLEMS WITH THE WATERFALL METHODOLOGY

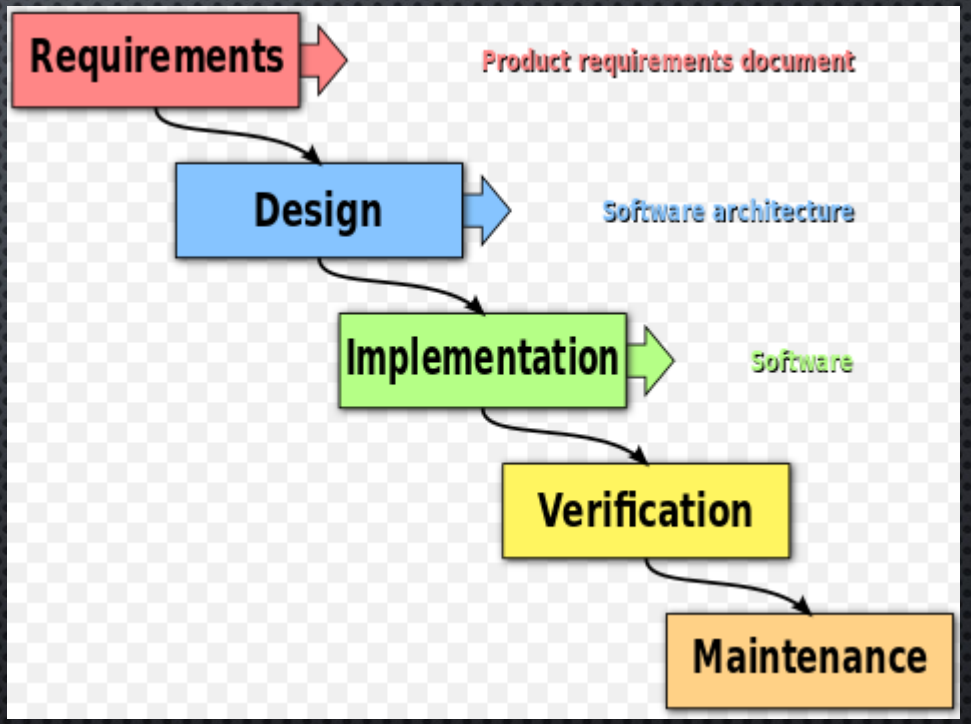
THE COMMON USAGE OF THE WATERFALL LIFECYCLE MODEL WAS SEQUENTIALLY FOLLOWING THE STEPS OF REQUIREMENTS, DESIGN, IMPLEMENTATION, VERIFICATION, AND MAINTENANCE.

1. DEFINE ALL REQUIREMENTS IN DETAIL UP-FRONT
2. DEFINE THE SYSTEM IN “TEXT” AND “DIAGRAMS”
3. IMPLEMENT THE SYSTEM “CODE, UNIT TEST, INTEGRATE”
4. INTEGRATE AND TEST THE SYSTEM COMPONENTS.

THIS MODEL DOES NOT WORK WELL WITH ADAPTING REQUIREMENTS.

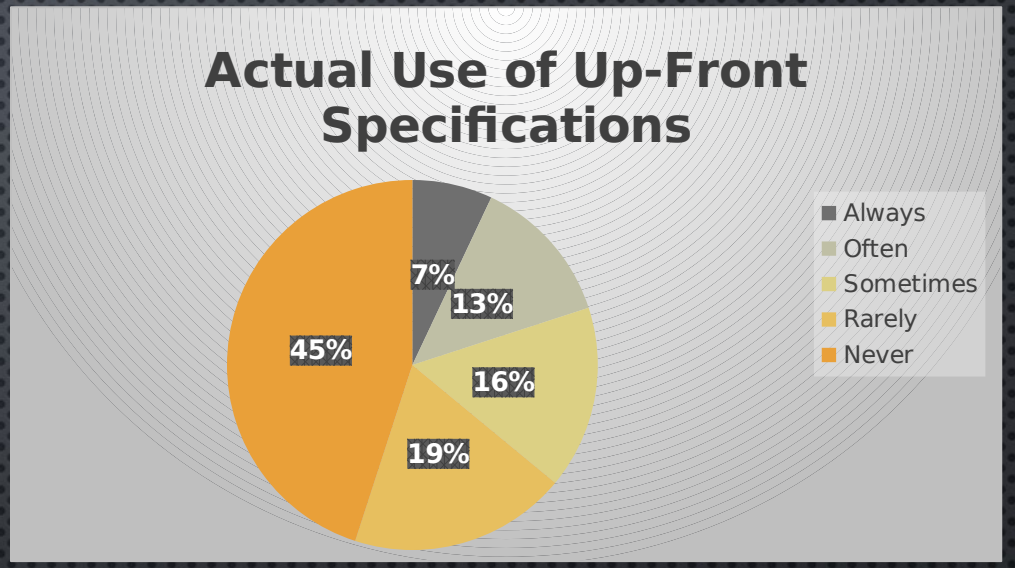
ALTHOUGH THIS WAS THE PREFERRED METHOD OF MANAGING A SOFTWARE PROJECT IN THE 1970S, TODAY'S RESEARCH CLEARLY SHOWS THAT THIS METHODOLOGY IS ASSOCIATED WITH HIGHER RISK, HIGHER FAILURE RATES, AND LOWER PRODUCTIVITY.

IN ADDITION, THE WATERFALL APPROACH RESULTS IN OVERWHELMING DEGREES OF COMPLEXITY SINCE IT DOESN’T BREAK THE DEVELOPMENT INTO MORE MANAGEABLE LEVELS OF COMPLEXITY (E.G., A SUBSET OF CAPABILITIES)



PROBLEMS WITH DEVELOPING UP-FRONT REQUIREMENTS

- IN ANOTHER STUDY THE AUTHOR STATES THAT UP-FRONT SPECIFICATION WITH A SIGN-OFF CAN NOT BE SUCCESSFULLY CREATED AND THAT A STUDY SHOWED THAT 45% OF THE FEATURES CREATED FROM EARLY SPECIFICATION WERE NEVER USED, WITH AN ADDITIONAL 19% RARELY USED AS SHOWN IN THE GRAPH ON THE RIGHT. [JOHNSON02]
- THE AUTHOR THEN PROCEEDED TO SAY "AVOID PREDICTIVE PLANNING BECAUSE YOU CAN NOT SIMPLY PLAN THE WORK AND WORK THE PLAN" WHEN DOING ITERATIVE SOFTWARE DEVELOPMENT.
- THIS WILL ONLY WORK IF YOU PROJECT IS NOT FIRM FIXED PRICE OR IF YOUR CUSTOMER HAS BOUGHT INTO THE IDEA OF YOU DELIVERING A SYSTEM WITH ONLY 75%-95% OF THE FEATURES THEY CONTRACTED!



AN AGILE CASE STUDY

AN AGILE PROJECT EXAMPLE - THE "STORY" OVERVIEW

IN CHAPTER 4 OF THE TEXT, THE AUTHOR PROVIDES AN EXCELLENT EXAMPLE OF USING A VARIETY OF AGILE TECHNIQUES (UP, EVO, SCRUM, AND XP) TO MANAGE A PROGRAM

- COMPANY: BORDER INFORMATION GROUP (BIG)
- PROJECT: BIOMETRIC RECORDING OR TRACKING HAZARDOUS EXTERNAL RADICALS (BROTHER)
- PROJECT MANAGER: CONVINCED UPPER MANAGEMENT THAT THE BEST WAY TO IMPLEMENT THIS PROJECT WAS TO USED TIMEBOXED ITERATIVE DEVELOPMENT COMBINED WITH TIMEBOXED INCREMENTAL DELIVER.
- IMPLEMENTATION TEAM: 1 PROJECT MANAGER, 1 SYSTEM ARCHITECT, 5 SOFTWARE DEVELOPERS
- *PROJECT START DATE = 1/1/2021…. 1ST TIMEBOXED INCREMENTAL DELIVERY (ID) TO CUSTOMER = 10/1/2021. DELIVERY DATA IS FIXED; OK FOR FEATURES TO FALL OUT OF 1ST DELIVERY TO CUSTOMER.*
  (REFER TO LECTURE 3, SLIDE 9 FOR DEFINITION OF ID)
- *THE CUSTOMER WILL BE AVAILABLE PART TIME EACH DAY. IN ADDITION, THERE WILL BE A DEDICATED SUBJECT MATTER EXPERT (SME) WHO'S PREVIOUS OCCUPATION OF BEING A BOARDER GUARD WILL BE AN ASSET TO THE TEAM.*
- *THE INITIAL SOFTWARE WILL BE DEPLOYED AT 2 LOW TRAFFIC AIRPORTS FOR 2 MONTHS TO GET THE BOARDER GUARD'S AND PASSENGER FEEDBACK ON SYSTEM.*

AN AGILE CASE STUDY

AN AGILE PROJECT EXAMPLE - THE “STORY” – WEEK 1

[SCRUM-01] TEAM RELOCATES TO A FACILITY AT 1 OF THE TARGET AIRPORTS THAT HAS A LARGE ROOM THAT COULD BE USED FOR COLLABORATION AND CUBICLES THAT CAN BE USED WHEN MEMBERS OF THE TEAM NEED QUIET TIME.

[SCRUM-02] TEAM TO PROVIDE A DEMO TO BIG’S UPPER MANAGEMENT EVERY 3-4 WEEKS.

[XP-01] CUSTOMER TO BE PRESENT EVERY MORNING, BOARDER GUARD TO PARTICIPATE AS SME FOR TEAM.

AN AGILE CASE STUDY

AN AGILE PROJECT EXAMPLE - THE "STORY" – WEEK 1 (CONTINUED)

[UP-01] TEAM TO HOLD A 2-DAY PLANNING AND REQUIREMENT WORKSHOP. GOAL IS TO BRAINSTORM REQUIREMENTS WHILE INCORPORATING A 20-PAGE WISH LIST FROM THE CUSTOMER.

- PROJECT MANAGER RECOMMENDS TEAM SELECT TOP 20% OF THE REQUIREMENTS AND CUSTOMER RECOMMENDATIONS BASED ON ARCHITECTURAL SIGNIFICANCE, RISK, AND VALUE. TEAM USED A DOT SYSTEM TO PRIORITIZE.
- TEAM SPENDS NEXT 2-DAYS ANALYZING REQUIREMENTS:
  - [UP-02] TEAM DECOMPOSED FUNCTIONAL REQUIREMENTS INTO MULTIPLE USE CASES
  - [EVO-01] TEAM IDENTIFIED NON-FUNCTIONAL (CUSTOMER) REQUIREMENTS THAT NEED TO BE QUANTIFIED (E.G., FAST RESPONSE) AND MEASURABLE (EASY TO USE) AS KEY REQUIREMENTS.
- TEAM LEAD SET EXPECTATIONS - FOR FIST ITERATIVE DEVELOPMENT CYCLE THAT WOULD START ON 01-09 AND END ON 01-26 WITH A DEMO CONSISTING OF A PARTIALLY RUNNING SYSTEM CONNECTED TO A BIOMETRIC METER.
  - [XP-02] TEAM DECIDES WHAT THEY CAN ACCOMPLISH WITH THE NEXT TWO WEEKS FROM THE 20% OF THE IDENTIFIED REQUIREMENTS
  - [UP-03] TEAM DECIDES TO IMPLEMENT A THE "POSITIVE PATH" ON A FEATURE THAT WILL TOUCH ON VARIOUS ARCHITECTURAL FEATURES OF THE SYSTEM.
  - [XP-03] TEAM DETERMINES THE NUMBER OF HOURS NEEDED TO IMPLEMENT THE WORK AND COMPARES IT TO THE NUMBER OF AVAILABLE HOURS WITHIN THE TIME BOX (ASSUMING NO OVERTIME). THE TEAM REDUCES THE SCOPE WITHIN THIS ITERATION TO FIT WITHIN THE TIMEBOX.
  - [SCRUM-03] PROGRAM MANAGER ENTERS FEATURES TARGETED FOR 1ST ITERATION INTO A SCRUM SPRINT BACKLOG SHEET.

AGILE CASE STUDY

AN AGILE PROJECT EXAMPLE - THE "STORY" – WEEK 2

[SCRUM-04] TEAM HOLDS DAILY 20 MIN. STAND-UP MEETINGS: REVIEWS GOAL FOR ITERATION, REMAINING TASKS WITHIN ITERATION, HOLDS TEAM Q&A, ASKS TEAM MEMBERS TO VOLUNTEER FOR ONE OF THE REMAINING TASKS TO COMPLETE.

[UP-04] CHIEF ARCHITECT EDUCATIONS TEAM ON POTENTIAL ISSUES AND DESIGN AND EXPLAINS THEIR VISION SO THE SYSTEM CAN BE DECOMPOSED INTO COMPONENTS. TEAM REFINES IDEA AND EXPLORES AND COORDINATES THE DESIGN IDEAS ON WHITE BOARD.

[XP-04] TEAM MOVES OUT ON CODING AFTER DECIDING TO USE XP PRACTICE OF TEST-DRIVEN DEVELOPMENT. ONE OF THE DEVELOPERS IS ASSIGNED THE TASK OF DEVELOPING ACCEPTANCE TEST. AS CLASSES AND UNIT TESTS ARE CREATED, THEY ARE CHECKED INTO A BUILD MACHINE THAT RUNS THE TESTS AS PART OF CONTINUOUS INTEGRATION WHICH RESULT IN PROBLEMS BEING QUICKLY IDENTIFIED AND RESOLVED.

[XP-05] EACH MORNING A TEAM MEMBER COLLECTS METRICS ON EVERYONE'S PROGRESS AND UPDATES THE SPRINT BACKLOG SPREADSHEET. COMPLETED TASKS ARE CROSSED OUT ON THE WHITE BOARD.

AGILE CASE STUDY

AN AGILE PROJECT EXAMPLE - THE "STORY" – WEEK 3+

AS CODE FROM MULTIPLE DEVELOPERS COME TOGETHER, THE TEAM BEGINS TO DEVELOP A SYNERGY AND THE OVERALL SYSTEM STARTS TO TAKE SHAPE AS PRODUCTION CODE AND UNIT TESTS ARE CHECKED IN DAILY.

AS THE TEAM APPROACHES THE TARGET DEMO DATE, THEY DO A CHECK OF THE BACKLOG AND DETERMINE IT THAT THEY HAVE ENOUGH TIME TO COMPLETE ALL OF THE ITEMS IN TIME FOR THE DEMO.

[SCRUM-05] TEAM HOLDS A DEMO TO THE BIG EXECUTIVES. EVEN THOUGH THE SYSTEM DOESN'T DO MUCH IT WAS IMPRESSIVE THAT THERE WAS A WORKING SYSTEM WITHIN 3 WEEKS.

THE BIG EXECUTIVE REQUEST THAT THE SYSTEM MUST ALSO INTERFACE WITH A 3RD PARTY FACE RECOGNITION SYSTEM BASED ON COMPETITIVE SYSTEMS CURRENTLY UNDER DEVELOPMENT.

THE TEAM BEINGS PLANNING ITS FEATURES OF THE SECOND ITERATION THAT WILL FOCUS ON THIS HIGH PRIORITY REQUEST.

AGILE CASE STUDY

WHERE WOULD YOU CONSIDER THIS EXAMPLE TO FALL WITHIN THE COCKBURN SCALE?

This classification model is used to identify methodologies best suited for UP, SCRUM, XP, and/or Evo process models.

L20 ?

E20 ?

D20 ?

Life-Critical

Company Fails

Lost Profits

Annoyance

REFERENCES

AGILE & ITERATIVE DEVELOPMENT, A MANAGER'S GUIDE, CRAIG LARMAN, EIGHTH EDITION, ADDISON WESLEY, NEW YORK, NY, COPYRIGHT 2004 BY PEARSON EDUCATION , INC.

SPECIFIC SOURCES THE AUTHOR QUOTED:

[JOHNSON02] - JOHNSON, J. 2002. KEYNOTE SPEECH, XP 2002, SARDINIA, ITALY

[JONES97] - JONES, C. 1997. APPLIED SOFTWARE MEASUREMENTS. MCGRAW HILL.

[PARKINSON58] - PARKINSON, N. 1958. PARKINSON'S LAW: THE PURSUIT OF PROGRESS. JOHN MURRAY.

[STANDISH94] - JIM JOHNSON, ET. AL 1994. CHAOS: CHARTING THE SEAS OF INFORMATION TECHNOLOGY. PUBLISHED REPORT. THE STANDISH GROUP

[THOMAS01] - THOMAS, M. 2001. "IT PROJECTS SINK OR SWIM" BRITISH COMPUTER SOCIETY REVIEW.