PROCESS

AGILE & ITERATIVE DEVELOPMENT (CHAPTER 2)

A MANAGER'S GUIDE BY: CRAIG LARMAN

WEEK 3: ITERATIVE & EVOLUTIONARY METHODS

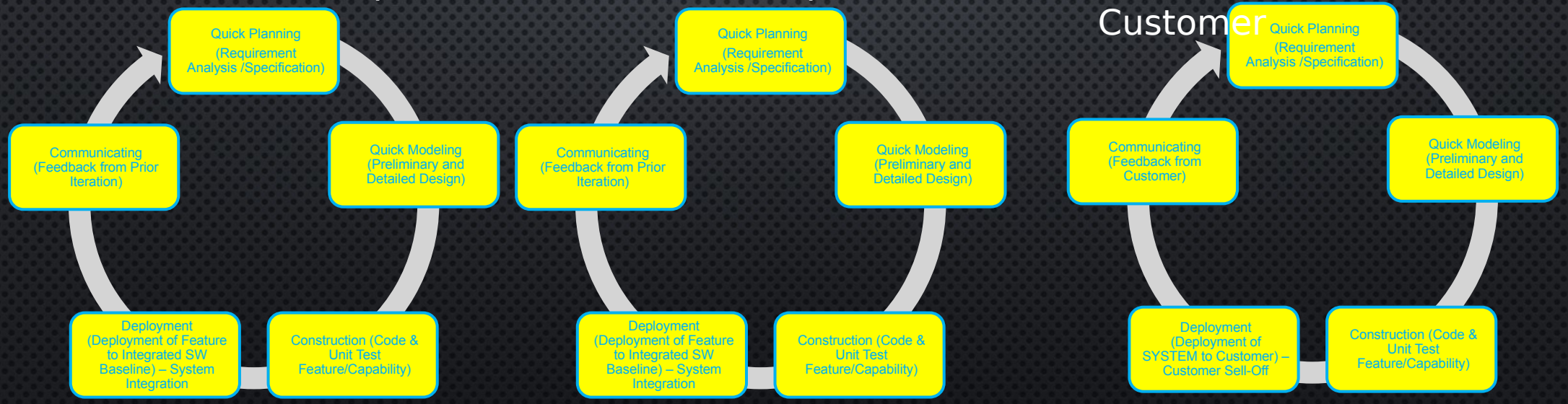
BY: JOSEPH MARTINAZZI

ITERATIVE (INCREMENTAL) DEVELOPMENT

ITERATIVE DEVELOPMENT LIFECYCLE MODEL

- AN ITERATIVE APPROACH TO DEVELOPING SOFTWARE IS WHERE REQUIREMENT ANALYSIS/SPECIFICATION, DESIGN, CODE & TEST OCCUR IN MULTIPLE ITERATIONS OVER THE LIFECYCLE OF THE PROGRAM.

Build 1 – FMK → Build 2 – Capabilities → Build 3 – Capabilities → Build X – Release to Customer



System Capabilities grow incrementally

ITERATIVE (INCREMENTAL) DEVELOPMENT

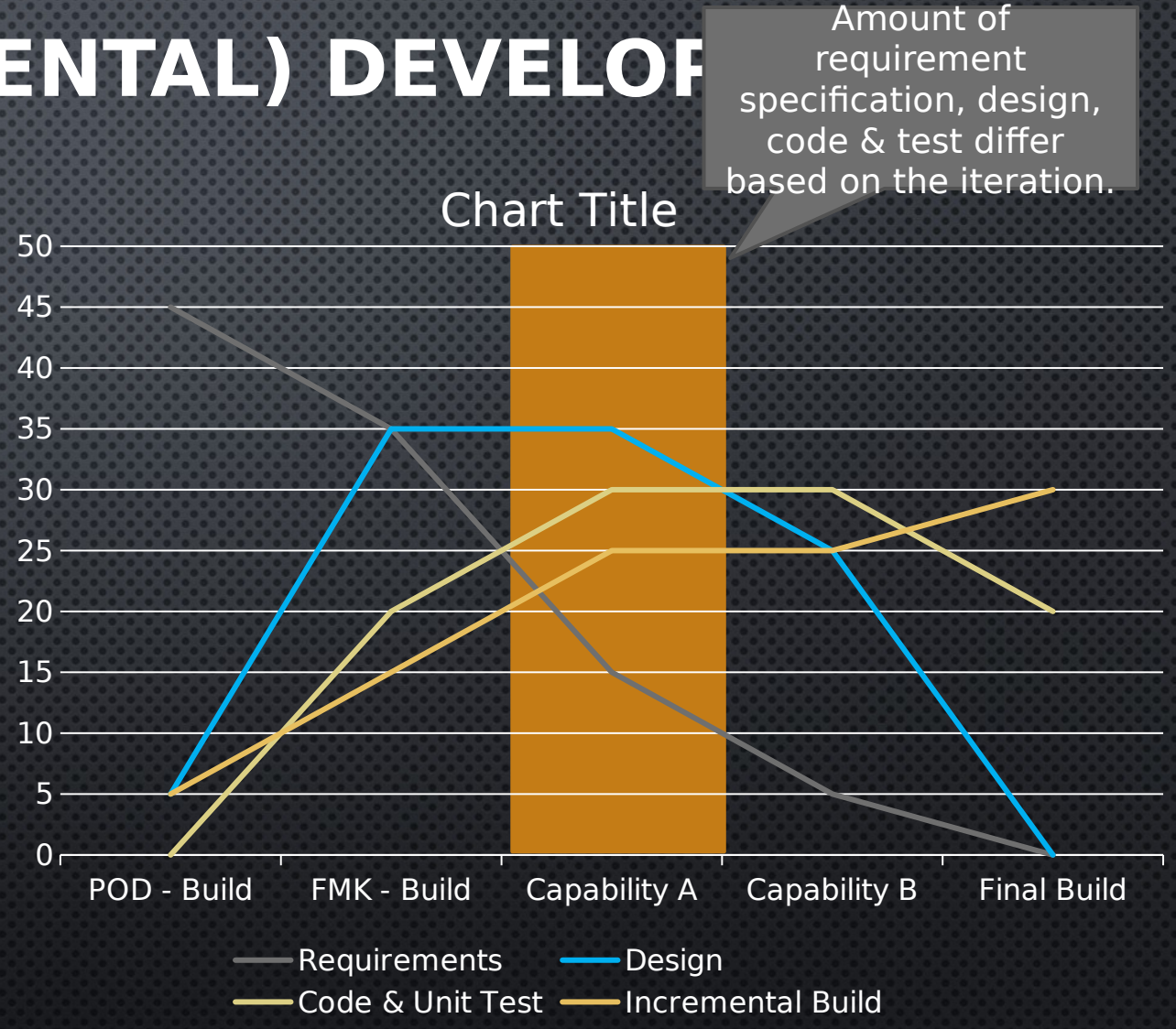
ITERATIVE DEVELOPMENT LIFECYCLE MODEL

- THE GOAL OF EACH ITERATION IS TO DELIVER TESTED CODE CONTAINING CAPABILITIES/FEATURES THAT ARE BUILT UP SEQUENTIALLY IN AN INTEGRATED SOFTWARE BASELINE OVER TIME.
- *THIS APPROACH ENABLES:*
  - *MULTIPLE TEAMS OF SOFTWARE DEVELOPERS TO CONTRIBUTE TESTED CODE TO AN INTEGRATED AND PARTIALLY TESTED STABLE BASELINE.*
  - *RUN REGRESSION TEST AGAINST THE BASELINE TO ENSURE BREAKAGE DOESN'T OCCUR.*
  - *TEST NEW CAPABILITIES/FEATURES ADDED TO THE BASELINE IN A CONTROLLED ENVIRONMENT.*
  - *DELIVER THE BASELINE TO OTHER TEAMS ON THE PROGRAM TO PERFORM INTERNAL TESTING OF REQUIREMENTS AND PERFORMANCE.*
  - *ENABLE FEATURES TO INCREMENTALLY BE TESTED BETWEEN MULTI-SUBSYSTEMS ON A PROGRAM.*

ITERATIVE (INCREMENTAL) DEVELOP

ITERATIVE DEVELOPMENT

- REQUIREMENTS NEEDED TO SUPPORT THE ITERATION ARE “FROZEN” PRIOR TO THE ITERATION STARTS
- PROJECTS TYPICALLY HAVE AT LEAST THREE INTERNAL ITERATIONS PRIOR TO THE FINAL ITERATION THAT IS RELEASED TO THE CUSTOMER.
- ITERATIONS CAN LAST FROM ONE WEEK IN DURATION TO SIX MONTHS IN DURATION ON PROJECTS THAT SPAN MULTIPLE YEARS. NOTE: THE AUTHOR STATES THAT THE RECOMMEND LENGTH OF AN ITERATION IS BETWEEN 1-6 WEEKS IN MODERN ITERATIVE METHODS.
- FEATURES/CAPABILITIES THAT DO NOT HAVE A DEPENDENCY ON A PRIOR ITERATION CAN EXECUTE IN PARALLEL.



The author compares an iteration to a self-contained mini-project containing production-quality capabilities.

ITERATIVE PLANNING

RISK-DRIVEN ITERATIVE DEVELOPMENT

A RISK DRIVEN APPROACH TO ITERATIVE DEVELOPMENT IS BASED ON:

IDENTIFYING THE MOST CHALLENGING OR RISKY REQUIREMENTS TO IMPLEMENT IN EARLY ITERATIONS.

THESE REQUIREMENTS TYPICALLY INVOLVE INCORPORATING NEW TECHNOLOGY, PERFORMANCE REQUIREMENTS, OR OTHER RISKS IDENTIFIED DURING THE INITIAL PLANNING STAGE OF THE PROGRAM.

EXAMPLE: AN EXAMPLE PROVIDED BY THE AUTHOR INVOLVED 2 REQUIREMENTS FOR A SYSTEM:

1. THE WEBPAGES TO BE GREEN AND
2. THE SYSTEM SHALL BE ABLE TO HANDLE 5,000 SIMULTANEOUS TRANSACTIONS (WHICH SHOULD BE IMPLEMENTED 1ST)

CLIENT-DRIVEN ITERATIVE DEVELOPMENT

A CLIENT DRIVEN APPROACH TO ITERATIVE DEVELOPMENT IS BASED ON THE CUSTOMER DEFINING THE FEATURES/CAPABILITIES CONTAINED IN THE NEXT ITERATION.

ADVANTAGE OF THIS APPROACH INCLUDE:

- CUSTOMER PRIORITIZES THE FEATURES/CAPABILITIES THAT ARE THE MOST IMPORTANT TO THEM FOR EARLY ITERATIONS.
- THE APPROACH CAN BE ADAPTIVE FOR EACH ITERATION BASED ON INSIGHT THE CUSTOMER GAINS DURING THE PREVIOUS ITERATIVE DEVELOPMENT CYCLE.

TIMEBOXED ITERATIVE DEVELOPMENT

TIMEBOXING

- IS AN APPROACH IN WHICH THE ITERATION RELEASE DATE IS FIXED.
- THIS APPROACH CAN APPLY TO ONE OR ALL ITERATIONS WITHIN A PROGRAM, HOWEVER THE TIMEBOX LENGTH FOR EACH ITERATION DOES NOT NEED TO BE EQUAL.
- IF FEATURES/CAPABILITIES ASSOCIATED WITH THE ITERATION CAN NOT BE MET WITHIN THE SCHEDULED COMPLETION DATE OF THE ITERATION (TIMEBOX), THEN THE FUNCTIONALITY WITHIN THE ITERATION IS REDUCED (SCOPE MOVED TO PROGRAM BACKLOG).
- MOST ITERATIVE INCREMENTAL DEVELOPMENT (IID) METHODS RECOMMEND TIMEBOXING OF 1-6 WEEKS IN DURATION.
- THREE-MONTH TO SIX-MONTH TIMEBOXING HAS BEEN SUCCESSFULLY EXECUTED ON LARGE PROGRAMS CONTAINING HUNDRED OF SOFTWARE DEVELOPERS!

TIMEBOXED ITERATIVE DEVELOPMENT

THE PROBABILITY THAT AN ITERATIVE DEVELOPMENT APPROACH WILL BE SUCCESSFUL IS DETERMINED BY FOUR VARIABLES THAT IMPACT A PROGRAM: TIME, SCOPE, RESOURCES (STAFF AND LAB EQUIPMENT), QUALITY.

- TIMEBOXING REMOVES THE VARIABLE OF TIME.
- PROCESS REMOVES THE VARIABLE OF QUALITY.
- THE IPTL/SPM NEEDS TO PREVENT EXTERNAL STAKEHOLDERS FROM CHANGING EITHER THE SCOPE OR THE RESOURCES ALLOCATED TO THE ITERATION ONCE IT BEGINS.
- THE TEAM CAN DE-SCOPE A TASK IF IT CAN NOT FIT WITHIN AN ITERATION'S TIMEBOX ONLY WITH THE APPROVAL OF THE IPTL/SPM.

  ALTHOUGH TIMEBOXING SHOULD NOT BE USED TO HAVE SOFTWARE DEVELOPERS WORK LONGER HOURS TO HIT A PROJECT DEADLINE, IT MAY BE NECESSARY IN SOME CASES BASED ON THE IMPACT TO THE OVERALL PROGRAM.

  ALTHOUGH IT IS OK TO PUSH FEATURES/CAPABILITIES TO THE NEXT ITERATION OR TO THE PRODUCT BACKLOG, IT IS NECESSARY TO UNDERSTAND

  - TECHNICAL DEPENDENCIES (IF THIS FEATURE IS MOVED TO A LATER ITERATION, WHAT SCHEDULE/COST IMPACT WILL THIS HAVE ON OTHER FEATURES/CAPABILITIES BEING DEVELOPED IN FUTURE INCREMENTS),
  - SCHEDULE DEPENDENCIES (BY MOVING THIS FEATURE TO EITHER A LATER SPRINT OR TO THE PRODUCT BACKLOG, WILL THERE BE A RESOURCE CONFLICT ON A LATER DEVELOPMENT ACTIVITY), AND
  - COST DEPENDENCIES (WILL THE TEAM COMPLETE THE CURRENT ITERATION WITHIN COST, OR ARE THEY PUSHING A COST-OVER RUN TO THE FUTURE). *FIRM-FIXED PRICE (FFP) PROGRAM VS. COST PLUS PROGRAM*

EVOLUTIONARY AND ADAPTIVE DEVELOPMENT

EVOLUTIONARY ITERATIVE DEVELOPMENT

IMPLIES THAT PROGRAM REQUIREMENTS, ESTIMATES, AND SOLUTIONS EVOLVE OR ARE REFINED OVER TIME VS. HAVING ALL REQUIREMENTS DEVELOPED UP FRONT AND FROZEN THROUGH THE COURSE OF THE ITERATIVE DEVELOPMENT LIFECYCLE.

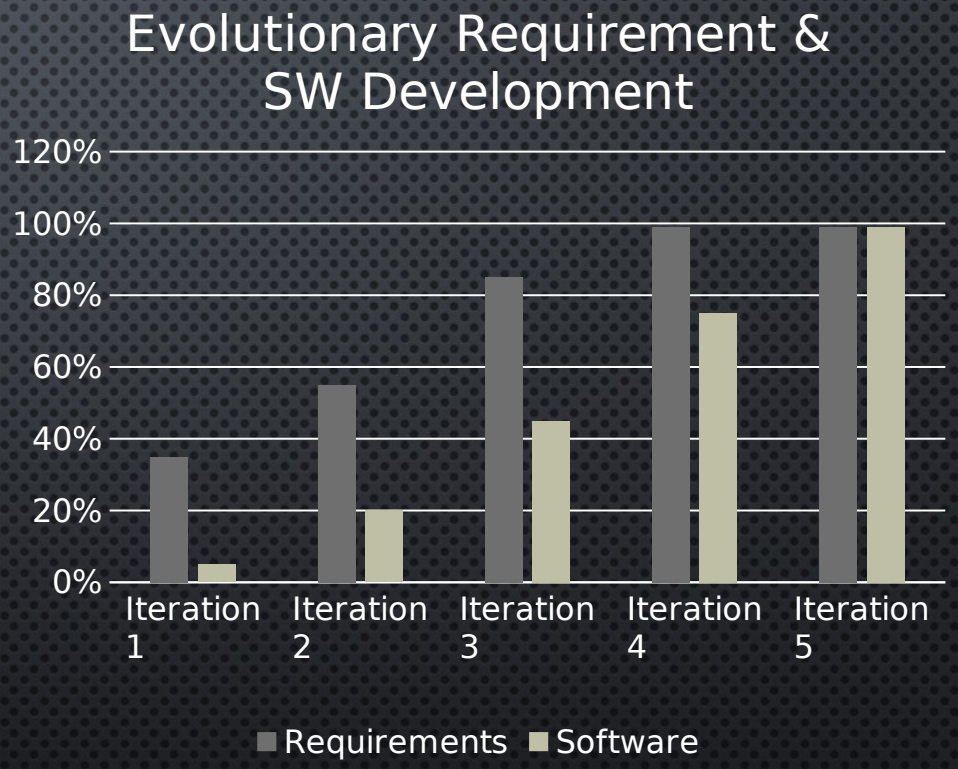
- FOCUS IS ON HIGH RISK, PERFORMANCE AND USABILITY REQUIREMENTS UP FRONT.
- RECOMMENDATION IS TO HAVE WORKSHOPS INCLUDING SYSTEMS AND SOFTWARE ENGINEERS TO FLUSH OUT THE DETAILS.

ADAPTIVE DEVELOPMENT

IMPLIES THAT DEVELOPMENT IS ADAPTED FOR EACH ITERATION BASED ON FEEDBACK OR INSIGHT GAINED DURING THE PREVIOUS ITERATIVE DEVELOPMENT CYCLE.

DOESN'T IMPLY THAT THERE ARE NO COST AND NO SCHEDULE BOUNDARIES, JUST THAT IT IS HARDER TO PREDICT UP FRONT AND CAN BE BETTER REVISED IN FURTHER ITERATIONS.

ROLLING WAVE PLANNING IS A WAY TO ACCOMPLISH THIS BY CREATING TASKS IN THE IMS THAT ARE DETAILED PLANNED AS WORK PACKAGES DURING THE NEXT 6 MONTHS OF THE PROGRAM AND THE REMAINING TASKS ARE PLANNED AT A HIGHER LEVEL IN A PLANNING PACKAGE.



INCREMENTAL AND EVOLUTIONARY DELIVERY

INCREMENTAL DELIVERY

IS THE PROCESS OF DELIVERING A SYSTEM TO THE CUSTOMER IN A SERIES OF EXPANDED CAPABILITIES/FEATURES. TIME BETWEEN AN INCREMENTAL DELIVERY CAN RANGE BETWEEN 3-12 MONTHS.

INCREMENTAL PRODUCT DELIVERY IS NOT THE SAME AS ITERATIVE DEVELOPMENT. EACH INCREMENTAL DELIVERY CAN BE COMPOSED OF MULTIPLE ITERATIVE DEVELOPMENT CYCLES.

EVOLUTIONARY DELIVERY

IS SIMILAR TO INCREMENTAL DELIVERY EXCEPT IT CAPTURES CUSTOMER FEEDBACK AND PROVIDES THAT AS GUIDANCE INTO THE NEXT DELIVERY.

CASE STUDY (THE MOST COMMON MISTAKES?)

COMPANY X-Y-Z ACKNOWLEDGES THAT THEY HAVE CHOSEN TO USE AN ITERATIVE DEVELOPMENT METHODOLOGY SINCE THE WATERFALL LIFECYCLE MODEL IS NOT VERY SUCCESSFUL. HOWEVER, THEY STATE THAT THEY WILL NOT BEGIN SOFTWARE DEVELOPMENT UNTIL THEY COMPLETE THE USE CASE ANALYSIS, INITIAL IMP AND IMS, AND THE SYSTEM LEVEL REQUIREMENT SPECIFICATIONS.

ALTHOUGH THE AUTHOR STATES THAT THIS IS ONE OF THE MOST COMMON MISTAKES THAT NEW ITERATIVE AND AGILE METHOD ADOPTERS MAKE. I BELIEVE HIS STATEMENT IS TOO GENERIC AND SHOULD SPECIFICALLY FOCUS ON THE FACT THAT NOT ALL REQUIREMENT SPECIFICATIONS NEED TO BE COMPLETED PRIOR TO BEGINNING THE SOFTWARE DEVELOPMENT EFFORT.

AS DISCUSSED DURING LECTURE 2 – THERE ARE MANY ORGANIZATIONAL AND PROGRAMMATIC PROCESSES THAT DETERMINE YOUR ABILITY TO EFFECTIVELY ADAPT ITERATIVE/AGILE METHODOLOGY INTO THE SOFTWARE DEVELOPMENT MODEL USED ON A PROGRAM.

- LARGE PROGRAMS THAT SPAN MULTIPLE YEARS AND/OR INCLUDE MULTIPLE SUBSYSTEMS NEED SYSTEM LEVEL REQUIREMENTS TO BE WRITTEN TO A LEVEL THAT CAN BE ALLOCATED TO EACH SUB-SYSTEM. ESPECIALLY HIGH RISK, PERFORMANCE, USABILITY AND INTERFACE REQUIREMENTS PRIOR TO STARTING SOFTWARE DEVELOPMENT IN THOSE AREAS.
- A FIRM-FIXED PRICE (FFP) CONTRACT IS A BINDING AGREEMENT BETWEEN YOUR COMPANY AND YOUR CUSTOMER THAT YOU WILL DELIVER THE REQUIREMENTS SPECIFIED IN THE CONTRACT WITHIN A SPECIFIED SCHEDULE AND FOR A SPECIFIED COST. THE CONTRACT MAY EVEN SPECIFY THAT YOU WILL CONDUCT A PRELIMINARY DESIGN REVIEW (PDR) DURING MONTH 3 OF THE CONTRACT AND A DETAILED DESIGN REVIEW DURING MONTH 9 OF THE CONTRACT WHICH WILL ULTIMATELY DRIVE WHAT THE FOCUS IS DURING THE EARLY ITERATIONS OF THE DEVELOPMENT LIFECYCLE.
- IN ADDITION, CUSTOMER INVOLVEMENT DURING ITERATIVE DEVELOPMENT MAY NOT BE DESIRABLE IF THE CONTRACT IS FIRM-FIXED PRICE AND THERE IS NO CONTRACTUAL MECHANISM TO ELIMINATE LESS IMPORTANT REQUIREMENTS SPECIFIED IN THE CONTRACT DURING THE ITERATIVE DEVELOPMENT LIFECYCLE.
- REMEMBER THE ONLY WAY TO MANAGE THE PROGRAM SCHEDULE IS TO HAVE A DETAILED IMS THAT CRITICAL PATH ANALYSIS CAN BE USED TO IDENTIFY WHAT IS DRIVING THE PROGRAM SO YOU CAN MANAGE YOUR SCHEDULE BUFFER. ALSO DETAILED SOFTWARE PLANNING ENABLES YOU TO CREATE AN INITIAL COST BASIS AT THE BUILD OR ITERATION LEVEL THAT CAN BE REFINED AS YOU GO.

SPECIFIC ITERATIVE & EVOLUTIONARY METHODS

UNIFIED PROCESS (UP)

UP OR RATIONAL UNIFIED PROCESS IS THE MOST WIDELY USED ITERATIVE APPROACH ACROSS THOUSANDS OF ORGANIZATIONS WORLDWIDE.

IT WAS DEVELOPED IN THE MID-1990'S WITH INPUTS FROM MANY EXPERIENCED SYSTEM ARCHITECTS AND DESIGNERS.

ORGANIZATIONS THAT USE THE UP METHOD TYPICALLY FOCUS ON THE CORE ARCHITECTURE OF A SYSTEM AND HIGH-RISK AREAS OF DEVELOPMENT IN EARLY ITERATIONS TO MINIMIZE RISK TO THE PROGRAM.

EVOLUTIONARY (EVO)

EVOLUTIONARY PROCESS WAS DEVELOPED IN THE 1960'S WITH THE FOCUS BEING SHORT ITERATIONS OF 1-2 WEEKS IN DURATION

EVO USES ADAPTIVE PLANNING AND FOCUSES ON THE HIGHEST VALUE-TO-COST RATION ITEMS FIRST.

EVO ALSO PROMOTES THE USE OF UNAMBIGUOUS AND QUALITY REQUIREMENTS THAT CAN BE QUANTIFIED OR MEASURED (IN OTHER WORDS REQUIREMENTS MUST BE TESTABLE).

REFERENCES

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