PROCESS

AGILE & ITERATIVE DEVELOPMENT (CHAPTER 1)

A MANAGER’S GUIDE BY: CRAIG LARMAN

WEEK 1: INTRODUCTION - DEVELOPMENT PROCESSES, ISO 12207 AND CMMI

BY: JOSEPH MARTINAZZI

UBIQUITY OF COMPUTING

TODAY SOFTWARE HAS BECOME EMBEDDED INTO TECHNOLOGY INDIVIDUALS USE AS A PART OF THEIR DAILY LIVES.

THIS INCLUDES APPLICATIONS ON A COMPUTER THAT AN INDIVIDUAL CAN CHOOSE TO USE:

- USING WORD PROCESSING/SPREAD SHEET APPLICATIONS ON A COMPUTER,
- USING THE INTERNET FOR RESEARCH AND E-COMMERCE, AND
- PLAYING COMPUTER GAMES.

AS WELL AS EMBEDDED SOFTWARE IN MOST DEVICES THAT INDIVIDUALS DON'T HAVE A CHOICE TO USE:

- USING A SMART PHONE OR MOST OTHER CONSUMER ELECTRONIC DEVICES,
- DRIVING A CAR
- FLYING ON AN AIRPLANE,
- USING ELECTRICITY OR DRINKING WATER, AND
- USING MEDICAL EQUIPMENT TO DIAGNOSE/TREAT ILLNESSES.

UBIQUITY OF COMPUTING

AS A RESULT, THERE IS A LOT OF **RESPONSIBILITY** ON **COMPANIES AND INDIVIDUALS** THAT WORK AT THESE COMPANIES TO TAKE CONSUMER PRIVACY & SAFETY INTO ACCOUNT WHEN DEVELOPING THESE PRODUCTS.

A WAY FOR A COMPANY TO ACHIEVE THIS GOAL IS TO ESTABLISH A WELL-DEFINED PROCESS AND TO ENSURE ITS EMPLOYEES FOLLOW THAT PROCESS!

UNFORTUNATELY, SOFTWARE PROGRAMS HAVE BECOME MORE AND MORE COMPLEX AND THEIR USE MAY NOT ALWAYS BE POSSIBLE TO PREDICT.

- SOFTWARE IS DEVELOPED BY PEOPLE WHO THINK SEQUENTIALLY, WHO HAVE LIMITED DOMAIN KNOWLEDGE, AND WHO MAKE MISTAKES
- IMPOSSIBLE TO DETERMINE THE EFFECTS OF CONCURRENCY (MULTIPLE PROCESSES AND EXTERNAL INPUTS)
- INABILITY TO VERIFY & VALIDATE SYSTEMS IN AN ACCURATE TEST ENVIRONMENT
- ACCEPTANCE OF FAULT TOLERANT SYSTEMS

UBIQUITY OF COMPUTING – IN SMART PHONES

EXAMPLE #1: **SMART PHONES AND THE IMPACT THEY HAVE ON SOCIETY?**

- SMARTPHONES HAVE ENABLED MANY NEW WAYS FOR PEOPLE TO CONNECT WITH ONE ANOTHER OUTSIDE OF CONVERSATION INCLUDING:
    - FACETIMING,
    - TEXTING,
    - TAKING AND SHARING PICTURES,
    - ACCESSING EMAIL (OVER THE INTERNET), AND
    - ACCESSING SOCIAL MEDIA SITES (OVER THE INTERNET).

- SMARTPHONE APPS HAVE ENDLESS USES THAT ENABLE INDIVIDUALS TO STREAM VIDEOS, LISTEN TO MUSIC, AND EVEN GET A FAST PASS TICKET AT DISNEYLAND.

UBIQUITY OF COMPUTING – IN SMART PHONES

EXAMPLE #1: **SMART PHONES AND THE IMPACT THEY HAVE ON SOCIETY?**

- UNFORTUNATELY, SMARTPHONES HAVE CREATED MANY UNFORESEEN ISSUES IN SOCIETY AS WELL:
  - USE OF SMARTPHONES WHILE DRIVING INCREASE THE RISK OF AN ACCIDENT.
  - PEOPLE USE SMARTPHONES IN INAPPROPRIATE PLACES AND THE FACT THAT THEY HAVE CAMERAS AFFECTS OUR PRIVACY IN PUBLIC AND NON-PUBLIC PLACES.
  - RESEARCHERS ARE LEARNING AN ENORMOUS AMOUNT ABOUT OUR BEHAVIOR FROM HOW WE USE OUR SMARTPHONE. INVASION OF PRIVACY

UBIQUITY OF COMPUTING – IN CRITICAL SYSTEMS

EXAMPLE #2: **POORLY DESIGNED USER INTERFACES IN SAFETY CRITICAL SYSTEMS**

CASE STUDY #1 - ISSUES RESULTING FROM POORLY DESIGNED USER INTERFACES RESULTED IN THE CRASH OF AMERICAN AIRLINES FLIGHT 965 NEAR CALI, COLOMBIA.

- THE PILOT INTENDED TO LOCK THE AUTOPILOT ONTO A BEACON WHILE APPROACHING THE AIRPORT. AFTER ENTERING "R", THE COMPUTER SYSTEM DISPLAYED 6 BEACONS WITH "R". NORMALLY, THE CLOSET BEACON IS DISPLAYED AT THE TOP OF THE LIST.
- IN THIS CASE THE BEACON AT THE TOP OF THE LIST WAS 100 MILES AWAY RESULTING IN THE PLANE TURNING MORE THAN 90 DEGREES CRASHING INTO A MOUNTAIN, ALL 159 PEOPLE ON BOARD WERE KILLED.

WHY WAS THIS INCONSISTENCY NOT DISCOVERED DURING TESTING?

WAS THERE SOME LATEN ERROR THAT ONLY OCCURRED BASED ON A CERTAIN SET OF CIRCUMSTANCES OR WAS THERE A PROCESS PROBLEM?

UBIQUITY OF COMPUTING – IN CRITICAL SYSTEMS

EXAMPLE #2: **POORLY DESIGNED USER INTERFACES IN SAFETY CRITICAL SYSTEMS**

CASE STUDY #2 - ISSUES RESULTING FROM POORLY DESIGNED USER INTERFACES RESULTED IN ASIANA AIRLINES FLIGHT 214 CRASHING IN SAN FRANCISCO.

- THE PILOT DID NOT REALIZE THAT THE SPECIFIC AUTOPILOT MODE HE SELECTED DISENGAGED AN AUTO-THROTTLE FEATURE RESULTING IN THE PLANE'S SPEED DECREASING TOO RAPIDLY ON APPROACH TO THE AIRPORT.
- IN THIS CASE THE TAIL OF THE PLANE BROKE OFF KILLING 3 PASSENGERS AND INJURING THE REST OF THE PASSENGERS.

WHY WAS THE PILOT UNAWARE OF THIS FEATURE IN THIS AIRCRAFT?

WAS THERE AN ISSUE WITH THE TRAINING MATERIAL OR ASSOCIATED FLIGHT SIMULATOR?

UBIQUITY OF COMPUTING – IN CRITICAL SYSTEMS

EXAMPLE #3: **FAILURE TO CORRECTLY ESTABLISH SAFETY AS A KEY PART OF A PROCESS.**

PEOPLE MAKE DECISIONS BASED ON FACTS BUT TEND TO ERROR ON THE SIDE OF CAUTION IN ABSENCE OF A CONVINCING CASE FOR SAFETY.

CASE STUDY #1 - **THE SPACE SHUTTLE CHALLENGER WAS DESTROYED AS A RESULT OF A BLOW BY** (BREACH IN RUBBER GASKET THAT ENABLED BURNING GAS TO IGNITE THE ROCKET FUEL).

- NASA HAD ORIGINALLY HALTED SHUTTLE OPERATIONS UNTIL THE BLOW BY ISSUE COULD BE RESOLVED. HOWEVER, ONCE A SOLUTION WAS IDENTIFIED AND WENT INTO PRODUCTION, SHUTTLE LAUNCHES WERE PERMITTED TO CONTINUE. FAILURE TO MAKE A CONVINCING CASE FOR SAFETY
- THE NIGHT BEFORE THE SHUTTLE LAUNCH WAS THE COLDEST NIGHT ON RECORD (RUBBER GETS BRITTLE WHEN IT IS COLD). ENGINEERS ARGUED FOR A DELAY BECAUSE THEY KNEW THE COLD WEATHER POSED A SEVERE THREAT. HOWEVER, IN THE END THE LAUNCH WENT AHEAD AS SCHEDULED SINCE THEY COULD NOT ABSOLUTELY PROVE THAT THE SYSTEM WAS NOT SAFE UNDER THE CURRENT CONDITIONS. FAILURE TO MAKE A CONVINCING CASE FOR SAFETY
- THE SHUTTLE EXPLODED ON TAKEOFF KILLING A SCHOOLTEACHER, A SCIENTIST FROM HUGHES AIRCRAFT COMPANY, AND ALL THE ASTRONAUTS ON BOARD.

UBIQUITY OF COMPUTING – IN CRITICAL SYSTEMS

EXAMPLE #3: **FAILURE TO CORRECTLY ESTABLISH SAFETY AS A KEY PART OF A PROCESS.**

PEOPLE MAKE DECISIONS BASED ON FACTS BUT TEND TO ERROR ON THE SIDE OF CAUTION IN ABSENCE OF A CONVINCING CASE FOR SAFETY.

CASE STUDY #2 – **THE CHERNOBYL NUCLEAR DISASTER**

- WAS A RESULT OF THE PLANT OPERATORS NOT UNDERSTANDING THE RAMIFICATIONS OF HAVING THE PLANT ONLINE FOR TWO YEARS EVEN THOUGH THE THEY KNEW THE BACKUP SYSTEMS COULD NOT OPERATE FOR 60-75 SECONDS IN THE EVENT OF AN ELECTRICAL POWER FAILURE.
- THESE OPERATORS FELL UNDER THE TRAP OF THINKING IT WAS OK TO CONTINUE OPERATIONS BECAUSE THEY WERE ATTEMPTING TO SOLVE THE PROBLEM.

UBIQUITY OF COMPUTING – IN CRITICAL SYSTEMS

EXAMPLE #3: **FAILURE TO CORRECTLY ESTABLISH SAFETY AS A KEY PART OF A PROCESS.**

PEOPLE MAKE DECISIONS BASED ON FACTS BUT TEND TO ERROR ON THE SIDE OF CAUTION IN ABSENCE OF A CONVINCING CASE FOR SAFETY.

CASE STUDY #3 – **THE CRASHING OF 2 BOEING 800-MAX JET AIRLINERS**.

UNDERSTANDING THE PROBLEM YOU ARE TRYING TO SOLVE USING SOFTWARE IS EXTREMELY IMPORTANT PRIOR TO DESIGNING THE SOLUTION.

- THE DESIGNERS CHOSE TO REUSE AN EXISTING HARDWARE DESIGN VS. CREATING A NEW DESIGN TO SAVE TIME AND MONEY.
- AT SOME POINT IN THE DEVELOPMENT PROCESS, THE ENGINEERS MUST HAVE REALIZED THE PLANE HAD AN ISSUE DUE TO WEIGHT DISTRIBUTION AND DECIDED TO USE SOFTWARE TO CORRECT THE ISSUE OF THE NOSE OF PLANE POINTING DOWN DURING TAKE-OFF.

DO YOU THINK USING SOFTWARE TO CORRECT A HARDWARE ISSUE WITH THE PLANE WAS THE RIGHT WAY TO GO?

DO YOU THINK THEIR PROCESS HAD ADEQUATE CHECKS FOR SAFETY?

UBIQUITY OF COMPUTING – FUTURE TECHNOLOGY

EXAMPLE #4: **SELF-DRIVING CARS** – ARE THEY GOOD OR BAD FOR SOCIETY?

UNDERSTANDING THE PROBLEM YOU ARE TRYING TO SOLVE USING SOFTWARE IS EXTREMELY IMPORTANT PRIOR TO DESIGNING THE SOLUTION.

- WILL THEY SAVE MONEY, OR WILL ROAD SYSTEMS NEED TO EQUIP WITH ADDITIONAL SENSORS TO AID THE FULLY AUTOMATED VEHICLES?
- WILL THEY REDUCE TRAFFIC (BY PICKING UP MULTIPLE INDIVIDUALS ON THEIR WAY TO WORK) OR WILL THEY CAUSE MORE CONGESTION (EMPTY VEHICLES PICK UP 1 INDIVIDUAL)?
- WILL THEY SAVE LIVES (95% OF ACCIDENTS ARE CAUSED BY HUMAN ERROR) OR WILL THEY PUT THE OCCUPANTS AT RISK (COMPUTERS CAN BE HACKED).

IN CASES WHERE A CRASH IS UNAVOIDABLE; HOW WILL THE SOFTWARE DECIDE WHAT OR WHO TO HIT?

SHOULD THE SOFTWARE ALWAYS PRIORITIZE THE LIVES OF INDIVIDUALS WITHIN THE VEHICLE OR SHOULD ITS CRASH AVOIDANCE ALGORITHM BE BASED ON SAVING THE GREATEST NUMBER OF LIVES?

IMPORTANCE OF ORGANIZATIONAL PROCESSES

WHAT IS A DEVELOPMENT PROCESS?

- A DEVELOPMENT PROCESS OR PROCESSES ARE USED TO DEFINE A SYSTEMATIC, DISCIPLINED, AND QUANTIFIABLE APPROACH TO THE DEVELOPMENT, OPERATION, AND MAINTENANCE OF AN END-PRODUCT.

WHY IS FOLLOWING A PROCESS IMPORTANT?

- PROCESS LAYS THE FOUNDATION OF HOW AN ORGANIZATION DEVELOPS WORK PRODUCTS, ESTABLISHES MILESTONES, ENSURES QUALITY, ENSURES SAFETY (WHEN APPLICABLE) AND MANAGES CHANGE TO ENSURE THE END-PRODUCT THEY PRODUCE MEETS THE NEEDS OF THEIR CUSTOMER IN A TIMELY MANNER.
- PROCESSES ARE USED IN THE PLANNING, SPECIFICATION, DESIGN, IMPLEMENTATION, AND TEST OF SPECIFICATIONS, SOFTWARE, AND HARDWARE TO SUPPORT INTEGRATED PRODUCT TEAMS AND PROGRAMS GOALS.
- IN ADDITION, PROCESSES ARE USED TO MANAGE VARIOUS ASPECTS OF PROGRAMS INCLUDING PROJECT TRACKING (COST & SCHEDULE), RISK MANAGEMENT, WORK PRODUCT PREPARATION AND PRODUCTION, PRODUCT REUSABILITY AND SIZE MEASUREMENT, CONFIGURATION MANAGEMENT, QUALITY ASSURANCE, AND TECHNICAL REVIEW PERIODICITY AND CONTENT.
- HOW WELL AN ORGANIZATION FOLLOWS IT PROCESSES ACTUAL IMPACTS ITS ABILITY TO BID ON GOVERNMENT RELATED CONTRACTS.

CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

- CMMI IS A PROCESS MODEL DEVELOPED BY THE SOFTWARE ENGINEERING INSTITUTE AT CARNEGIE MELLON UNIVERSITY AS MECHANISM TO IMPROVE PERFORMANCE THROUGHOUT AN ORGANIZATION.

- THE GOAL OF THE MODEL IS FOR ORGANIZATIONS TO CREATE A SET OF BEST PRACTICES FOR RESOLVING PROCESS ISSUES, MINIMIZING PROGRAM RISKS, AND CREATING A QUALITY PRODUCT IN THE MOST EFFICIENT MANNER.

- ORGANIZATIONS THAT HAVE MASTERED THESE PRACTICES CAN BE ASSESSED BY THE CMMI INSTITUTE TO DETERMINE WHAT CMMI MATURITY LEVEL THEY ARE OPERATING AT.

- THE GOVERNMENT TYPICALLY REQUIRES AN ORGANIZATION TO HAVE A CMMI MATURITY LEVEL OF ANYWHERE FROM A 3 TO A 5 TO EVEN BE CONSIDERED AS A CANDIDATE TO BID OR WORK ON A US GOVERNMENT CONTRACT. ORGANIZATIONS THAT OBTAIN A CMMI MATURITY LEVEL OF 4 OR 5 ARE VIEWED TO BE MATURE.

CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

- **STANDARD CMMI APPRAISAL METHOD FOR PROCESS IMPROVEMENT (SCAMPI)** - IS THE OFFICIAL METHOD USED BY THE CMMI INSTITUTE TO EVALUATE AN ORGANIZATIONS LEVEL OF MATURITY. THERE ARE THREE APPRAISAL CLASSES: CLASS A, CLASS B, AND CLASS C.

  - CLASS A – SCAMPI A IS THE MOST COMPREHENSIVE OF THE APPRAISAL CLASSES AND RESULTS IN PROVIDING THE ORGANIZATION WITH AN OFFICIAL MATURITY LEVEL RATING.

  - CLASS B – SCAMPI B IS LESS A RIGOROUS APPRAISAL METHOD. THIS APPRAISAL IS USEFUL WHEN AN ORGANIZATION WANTS TO PERFORM AN INTERNAL SELF APPRAISAL TO DETERMINE WHAT MATURITY LEVEL THEY WOULD ACHIEVE IF A SCAMPI A AUDIT WAS CONDUCTED. THIS IS USED TO FIND POTENTIAL ISSUES AND CORRECT PROBLEMS PRIOR TO GOING THROUGH THE OFFICIAL SCAMPI A ASSESSMENT.

  - CLASS C – SCAMPI C IS A SHORT FLEXIBLE APPRAISAL METHOD THAT ASSISTS AN ORGANIZATION'S BEST PRACTICES AND HOW WELL THEY ALIGN WITH CMMI PRACTICES. IT CAN BE USED AT A HIGH-LEVEL TO ADDRESS ORGANIZATIONAL ISSUES OR AT A LOWER-LEVEL TO ADDRESS PROGRAM OR PROCESS ISSUES AND TO ADDRESS SPECIFIC RISK AREAS.

CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

CMMI MATURITY LEVELS

- **CMMI LEVEL 1: INITIAL** – PROCESSES WITHIN THE ORGANIZATION ARE NOT WELL DEFINED AND MAY NOT BE REPEATABLE. THE ORGANIZATION RELIES ON KEY INDIVIDUALS TO KEEP THINGS RUNNING AND IS MORE REACTIVE VS. PROACTIVE IN MANAGING PROJECTS. PROGRAMS TYPICALLY DO NOT GET COMPLETED WITHIN COST OR SCHEDULE DUE TO INEFFICIENCIES.
- **CMMI LEVEL 2: MANAGED AND REPEATABLE** – PROCESSES WITHIN THE ORGANIZATION ARE DEFINED AND PRODUCE REPEATABLE RESULTS. THE ORGANIZATION HAS ACHIEVED A BASIC LEVEL OF PROJECT MANAGEMENT IN WHICH PROGRAMS ARE PLANNED, REQUIREMENTS MANAGED, AND PROCESSES/WORK PRODUCTS ARE MONITORED, MEASURED, AND CONTROLLED.
- **CMMI LEVEL 3: DEFINED** – PROCESSES WITHIN THE ORGANIZATION ARE STANDARDIZED TO PROVIDE CONSISTENT RESULTS ACROSS PROGRAM EXECUTION. KEY PROGRAM AND TECHNICAL PROCESSES INCLUDE INTEGRATED PROGRAM MANAGEMENT, CONFIGURATION MANAGEMENT, REQUIREMENTS DEVELOPMENT, RISK MANAGEMENT, CAUSAL ANALYSIS AND RESOLUTION, DESIGN, TEST, INTEGRATION, VERIFICATION & VALIDATION AND TRAINING.
- **CMMI LEVEL 4: QUANTITATIVELY MANAGED** – PROCESSES WITHIN THE ORGANIZATION ARE MATURE ENOUGH THAT THEY CAN BE MEASURED USING DEFINED METRICS TO MINIMIZE PROGRAM RISKS AND CORRECT PROCESS DEFICIENCIES.
- **CMMI LEVEL 5: OPTIMIZING** – PROCESSES WITHIN THE ORGANIZATION ARE MATURE ENOUGH THAT THEY ARE BOTH STABLE AND FLEXIBLE ALLOWING FOR CONTINUOUS PROCESS IMPROVEMENT AS NEW TECHNOLOGY IS INCORPORATED INTO THEIR WORK PRODUCTS.

KEY MANAGEMENT AND DISCIPLINE PROCESSES

Integrated Program Management

Program Organization

Key Program Processes: 1. Program Schedule and Milestones, 2. Program Work Breakdown Structure/Cost Collection Method, 3. Program Requirements, 4. Program Verification & Validation Plan, 5. Program Change Management Plan, 6. Program Risk Management Plan, and 7. Program Stake Holder Involvement Plan, 8. Program Quality Assurance Plan

Systems & Hardware Engineering Disciplines

Systems Organization

Key Systems Processes: 1. System Requirement Specifications, 2. Software Requirements Specifications (SRS), 3. Systems Verification and Validation Plan.

Hardware Organization

Software Engineering Discipline

Software Organization

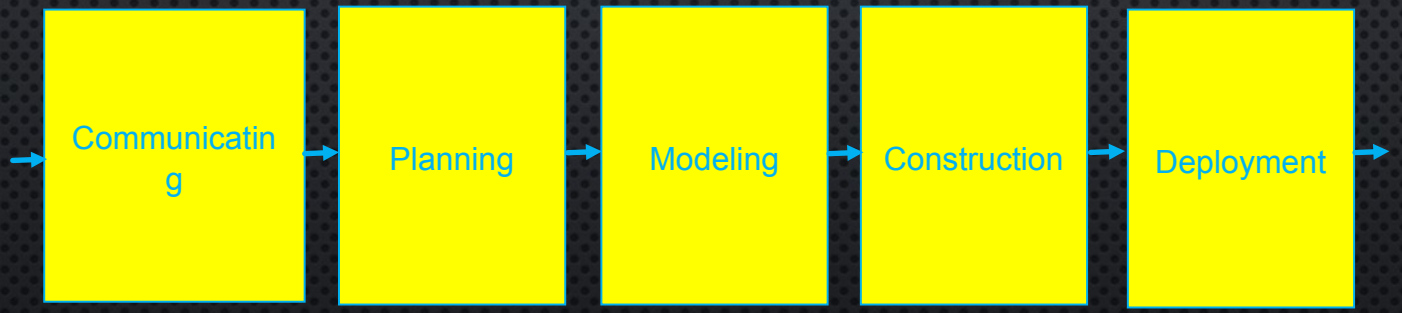
Key Software Processes: 1. Software Development Plan (SDP), 2. Software Build Plan (SBP), 3. Software Preliminary and Detail Design, 4. Software Code & Unit Test Plan, 5. Software Coding Standards (Language Specific), 6. Software Integration Plan, 7. Software Verification Plan, 8. Software Configuration Management Plan

PROCESS FLOW

WHAT IS A PROCESS FLOW?

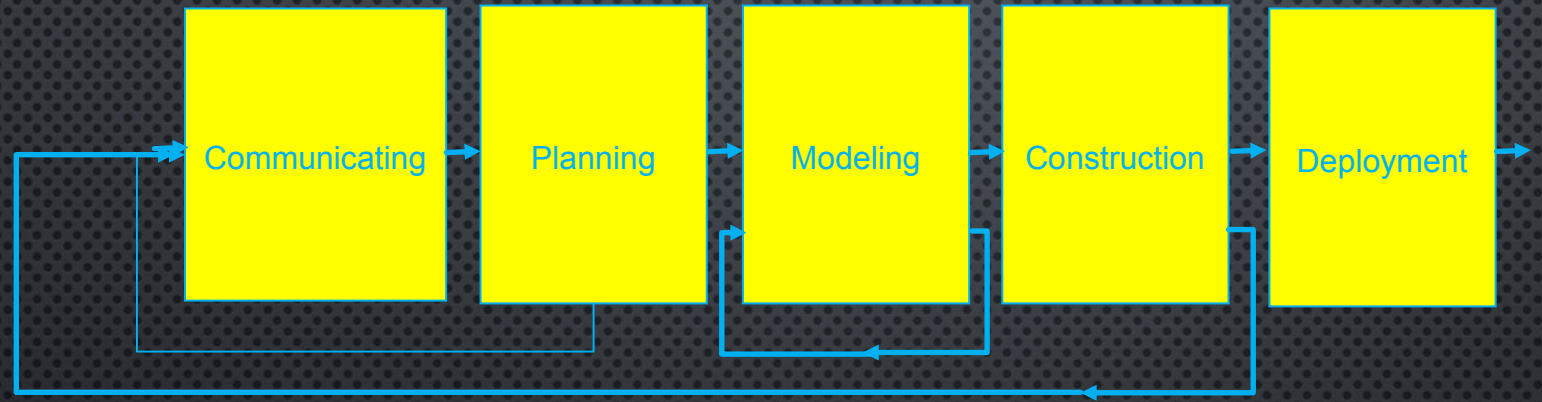
- A PROCESS TYPICALLY OCCURS WITHIN A GENERIC PROCESS FRAMEWORK OF COMMUNICATING, PLANNING, MODELING, CONSTRUCTION, AND DEPLOYMENT.
- A PROCESS FLOW CAN BE LINEAR, ITERATIVE, EVOLUTIONARY OR PARALLEL.

LINEAR PROCESS FLOW

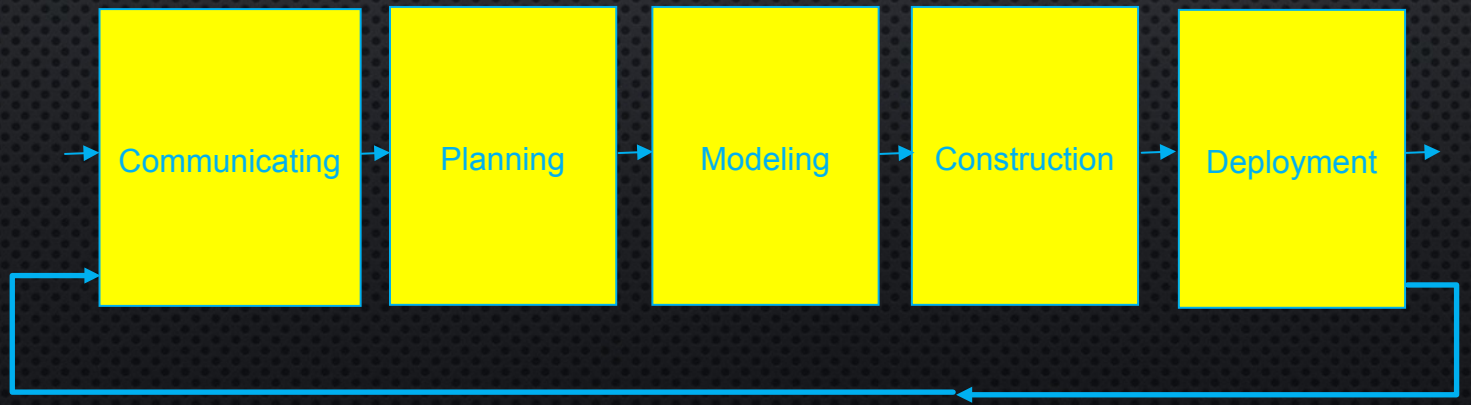


PROCESS FLOW

ITERATIVE PROCESS FLOW

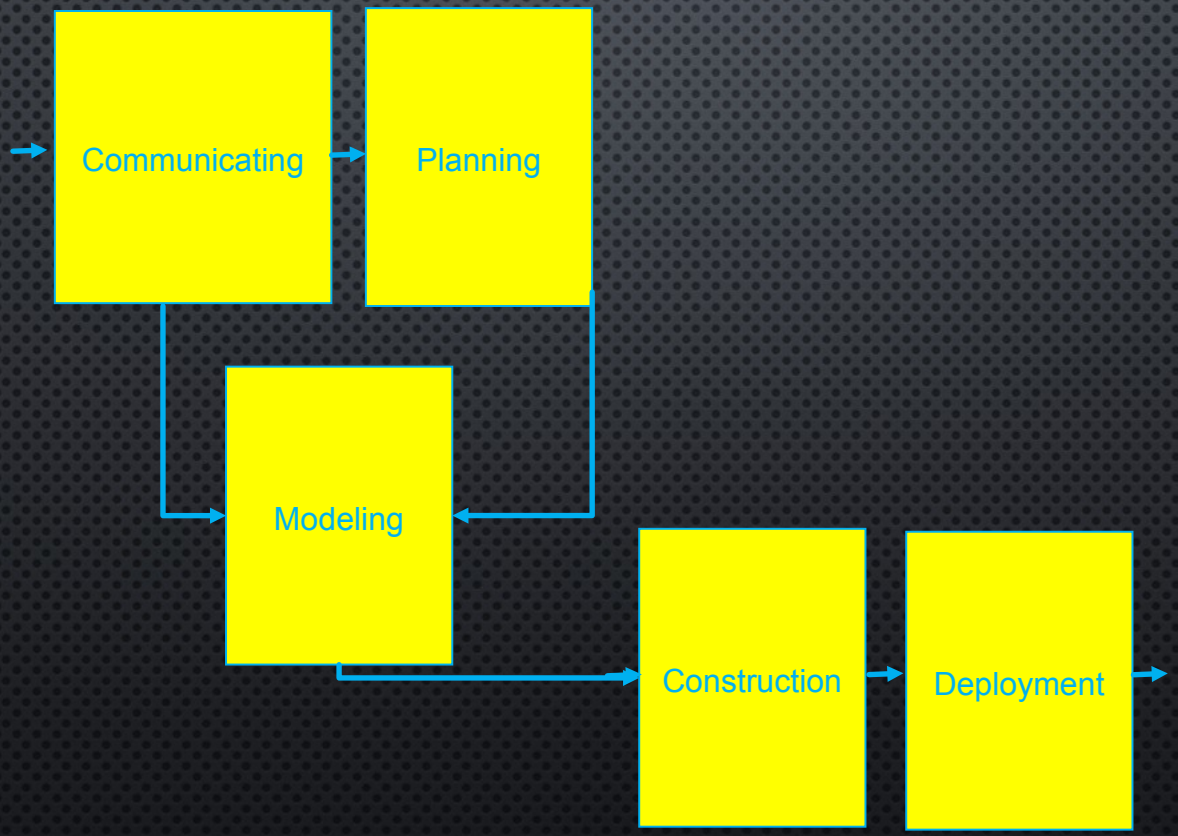


EVOLUTIONARY PROCESS FLOW



PROCESS FLOW

PARALLEL PROCESS FLOW



PROCESS MODEL

WHAT IS A PROCESS MODEL?

- A PROCESS MODEL IS USED TO DEFINE A SYSTEMATIC, DISCIPLINED, AND QUANTIFIABLE APPROACH TO THE DEVELOPMENT, OPERATION, AND MAINTENANCE OF A SOFTWARE WORK PRODUCT.
- PROCESS MODELS CAN BE PRESCRIPTIVE (IN WHICH TASKS ARE COMPLETED IN A SEQUENTIAL FASHION) OR INCREMENTAL (IN WHICH TASKS ARE COMPLETED IN LINEAR & PARALLEL FASHION) AND EVOLUTIONARY (IN WHICH TASKS ARE COMPLETED INCREMENTALLY WITH EACH INCREMENT PROVIDING MORE CAPABILITY).

PRESCRIPTIVE PROCESS MODELS

- WATERFALL LIFECYCLE MODEL – IS A SEQUENTIAL DEVELOPMENT MODEL. IT WAS THE PRIMARY MODEL USED IN DEPARTMENT OF DEFENSE (DOD) CONTRACTS FROM THE 1980'S-1990'S. UNFORTUNATELY, MANY OF THESE PROGRAMS FAILED TO PRODUCE THE END PRODUCT WITHIN COST AND SCHEDULE AND MANY FAILED TO PRODUCE AN END PRODUCT AT ALL.
- VERIFICATION & VALIDATION (V) MODEL – IS A SEQUENTIAL DEVELOPMENT MODEL.

PROCESS MODEL

ITERATIVE AND EVOLUTIONARY PROCESS MODELS

- **PROTOTYPING MODEL** – ALTHOUGH PROTOTYPING CAN BE USED AS A STAND-ALONE PROCESS MODEL, IT IS TYPICALLY USED IN SITUATIONS IN WHICH REQUIREMENTS AND/OR THE LOOK-AND-FEEL OF THE USER INTERFACE NEED ADDITIONAL INPUT FROM THE CUSTOMER.
- **BOEHM SPIRAL MODEL** – IS AN ITERATIVE AND EVOLUTIONARY DEVELOPMENT MODEL.
- **UNIFIED PROCESS (UP) MODEL** – IS AN ITERATIVE AND EVOLUTIONARY DEVELOPMENT MODEL.
- **SCRUM AGILE PROCESS MODEL** – IS AN ITERATIVE AND EVOLUTIONARY DEVELOPMENT MODEL.
- **EXTREME PROGRAMMING (XP) AGILE PROCESS MODEL** – IS AN ITERATIVE AND EVOLUTIONARY DEVELOPMENT MODEL.
- **EVOLUTIONARY (EVO) PROJECT MANAGEMENT PROCESS MODEL** – IS AN ITERATIVE AND EVOLUTIONARY DEVELOPMENT MODEL.

COMMONALITY AMONG PROCESS MODELS

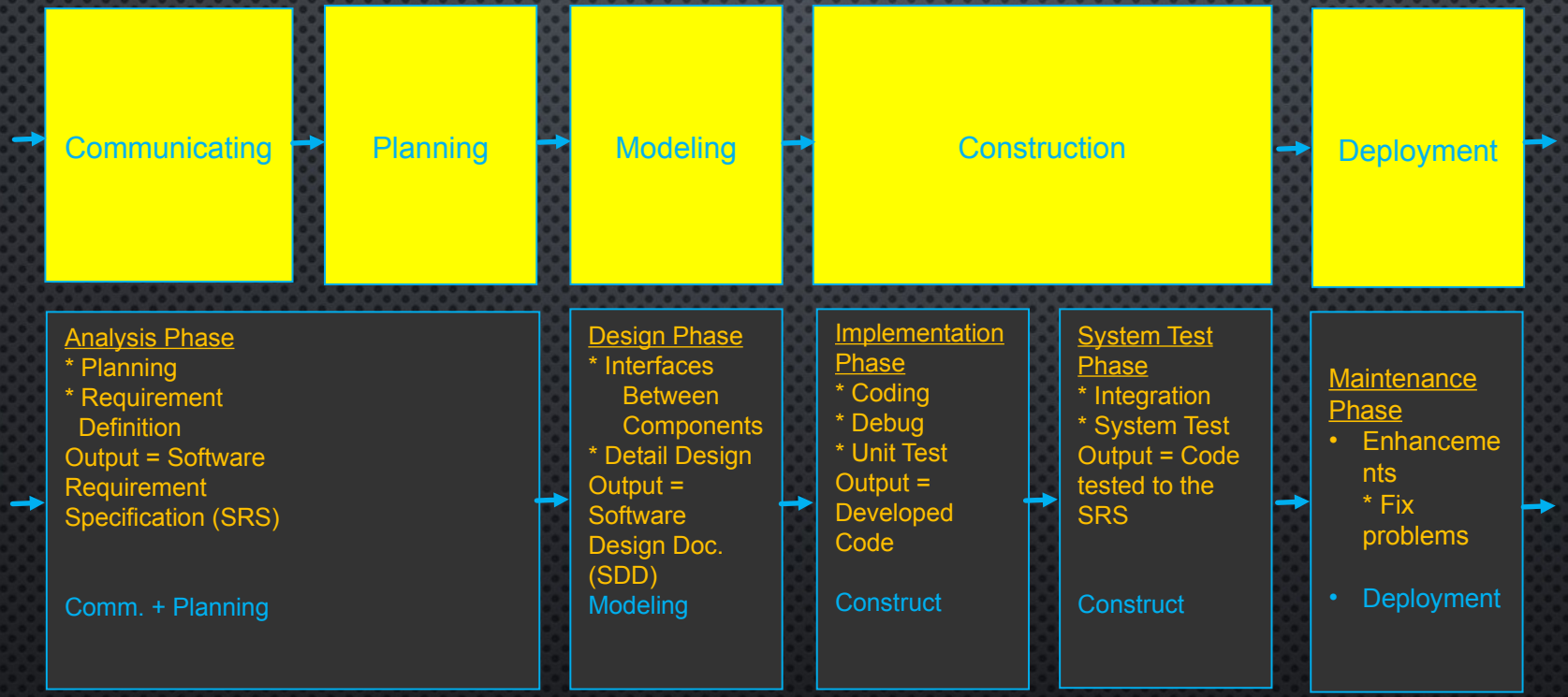
- **COMMUNICATING/PLANNING** – COMMUNICATING/COLLABORATING WITH THE CUSTOMER AND OTHER STAKEHOLDERS, DEFINING THE REQUIREMENTS (THIN SPECIFICATION), AND CREATING A SOFTWARE DEVELOPMENT PLAN DESCRIBING THE WORK, TECHNICAL TASKS, RISKS, RESOURCES, PROCESSES, AND WORK PRODUCTS TO BE PRODUCED WITHIN A SPECIFIC SCHEDULE AND COST ESTIMATE.
- **MODELING** – SELECTING THE BEST DESIGN METHODOLOGY TO PRODUCE A QUALITY WORK PRODUCT ON TIME AND WITHIN BUDGET.
- **CONSTRUCTION** – IMPLEMENTING THE SOLUTION (CODE AND TEST).
- **DEPLOYMENT** – PRODUCT (COMPLETED OR PARTIAL ITERATION) IS DELIVERED TO THE CUSTOMER/STAKEHOLDER WHO EVALUATES THE PRODUCT AND PROVIDES FEEDBACK.

COMMONALITY AMONG PROCESS ACTIVITIES

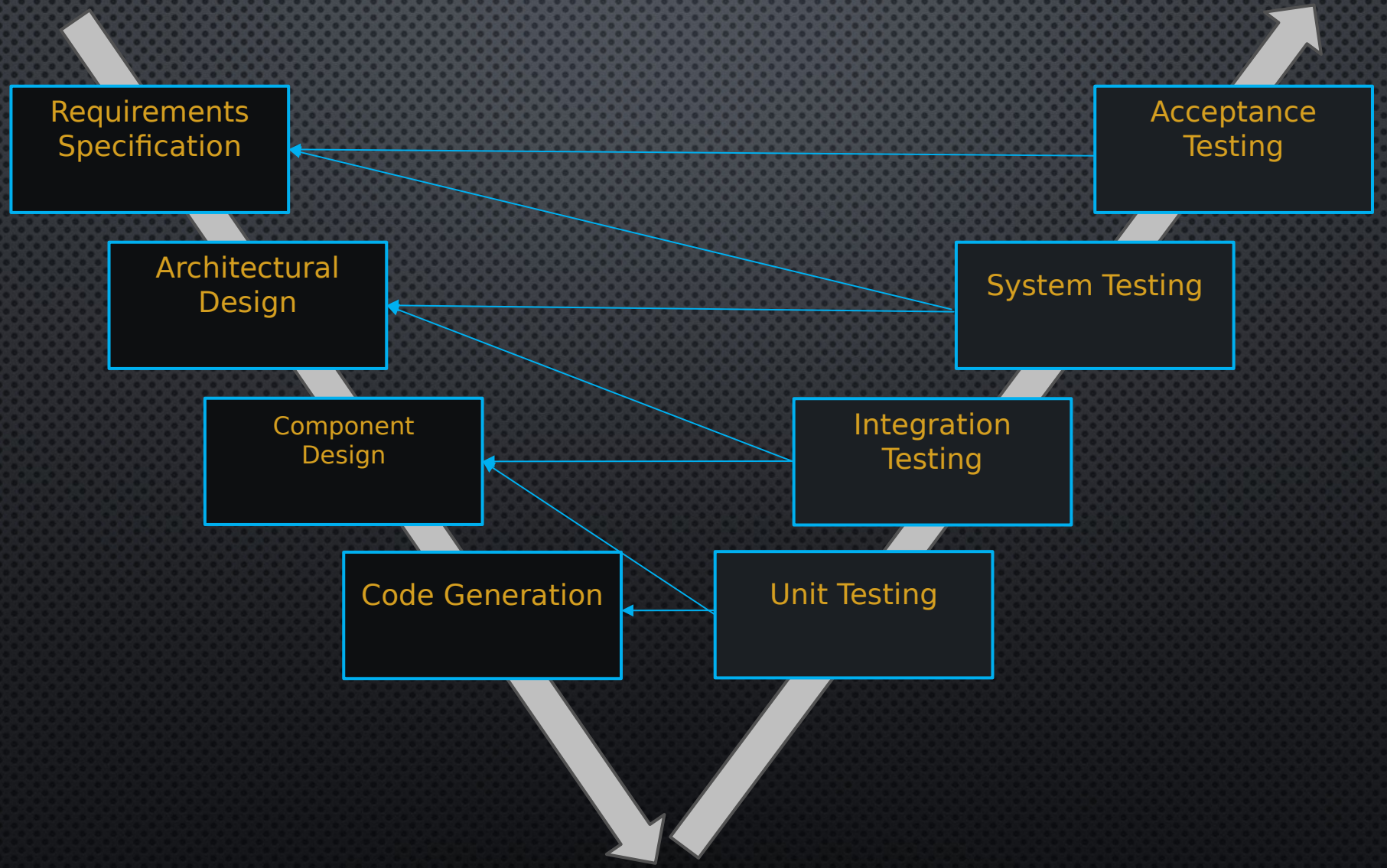
ALTHOUGH THE TERMINOLOGY BETWEEN PROCESS MODELS IS UNIQUE, THEY HAVE COMPARABLE ACTIVITIES AND PRODUCE SIMILAR WORK PRODUCTS.

12207-2017 ISO/IEC/IEEE Systems and SW Engineering – SW Life-cycle Processes	Waterfall Life-Cycle Model (Example)
System Requirements Analysis	Analysis – planning and requirements definition
System Architecture Design	Analysis – planning and requirements definition
Software Requirements Analysis	Analysis – planning and requirements definition
Software Architecture Design	Design – software component interface design
Software Detailed Design	Design – software component internal design
Software Coding and Test	Implementation – software component development and unit test
Software Integration	Software Test – software component integration
Software Qualification Testing	Software Test – software requirements verification
Software Installation	Maintenance – software deployment
Software Acceptance Testing	Maintenance – software system requirements verification and maintenance

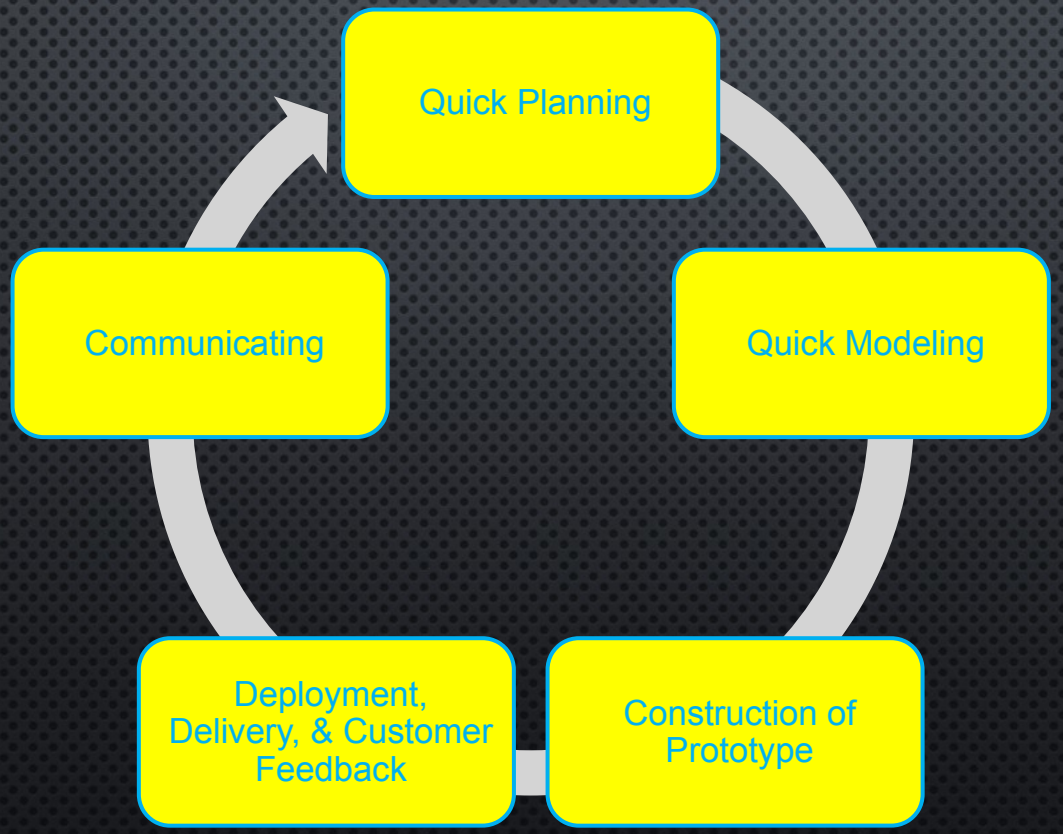
PRESCRIPTIVE PROCESS MODEL - WATERFALL



PRESCRIPTIVE PROCESS MODEL – V-MODEL

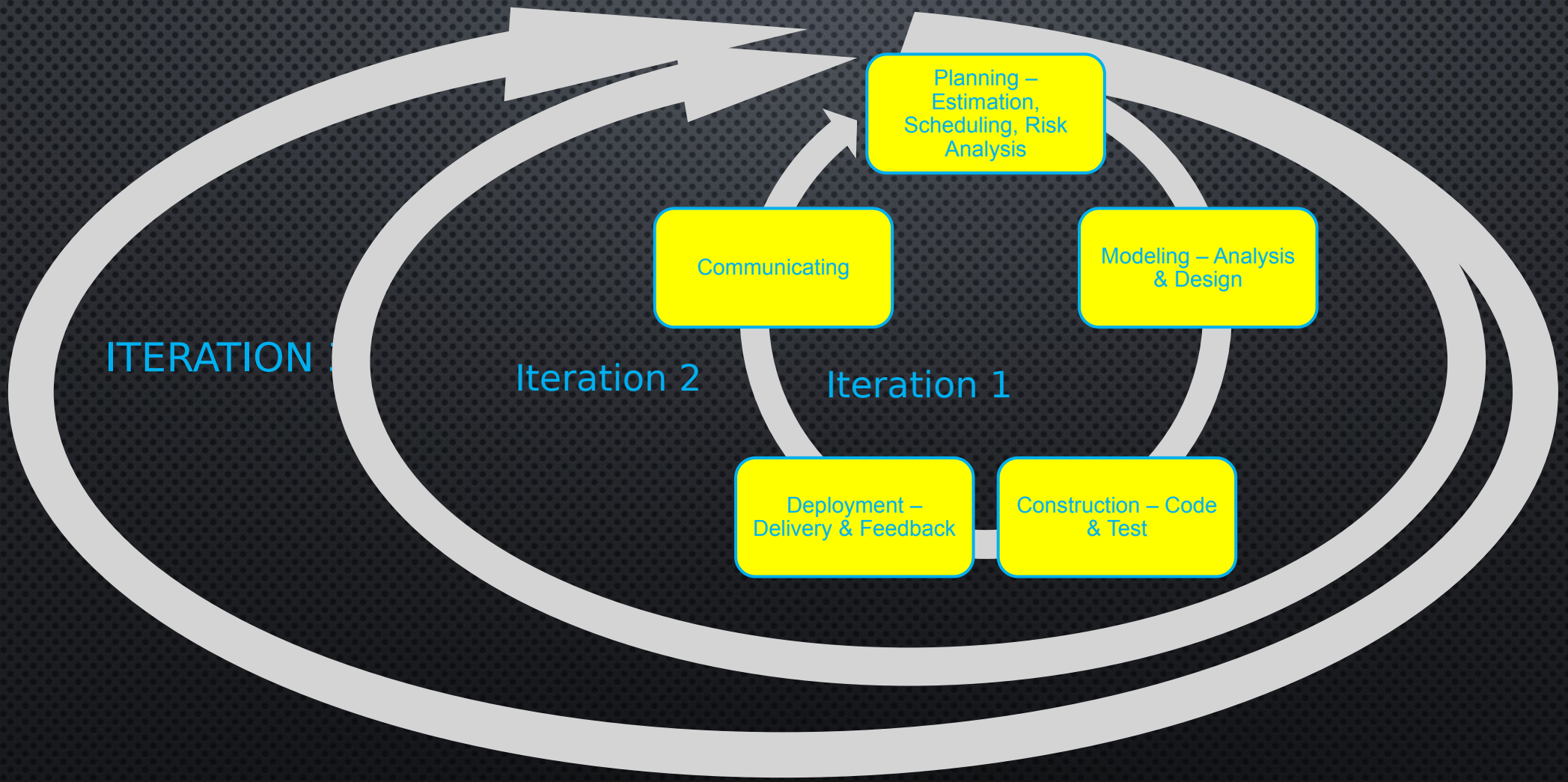


ITERATIVE AND EVOLUTIONARY PROCESS MODELS



ITERATIVE AND EVOLUTIONARY PROCESS MODELS

BOEHM SPIRAL MODEL



ITERATIVE AND EVOLUTIONARY PROCESS MODELS

UNIFIED PROCESS (UP) MODEL

SCRUM AGILE PROCESS MODEL

EXTREME PROGRAMMING (XP) AGILE PROCESS MODEL

EVOLUTIONARY (EVO) PROJECT MANAGEMENT PROCESS MODEL

NEW PRODUCT DEVELOPMENT

ACCORDING TO THE AUTHOR, THE "WATERFALL" LIFECYCLE MODEL IS MORE IN LINE WITH "PREDICTABLE MANUFACTURING" IN WHICH PROGRAM PLANNING AND REQUIREMENT SPECIFICATIONS OCCUR UP-FRONT AND IN WHICH ESTIMATES ARE BASED ON KNOWN METHODOLOGIES.

Predictable Manufacturing	New Product Development
Development effort and cost can be determined up front.	Not possible to estimate development effort and cost until empirical data is possible.
Ability to create a detailed schedule containing all the activities that need to be performed.	Ability to create a detailed schedule requires feedback from previous builds.
Adapting to unpredictable change is NOT the NORM. Change rates are relatively low.	Creative adaptation to unpredictable change is the NORM (e.g. Critical Chain, Agile Feedback meeting, etc.) Change rates are high.

NEW PRODUCT DEVELOPMENT

- SOFTWARE TYPICALLY FALLS INTO THE NEW PRODUCT DEVELOPMENT STAGE, ESPECIALLY WHEN NEW TECHNOLOGY IS USED. ITERATIVE AND AGILE METHODS TEND TO BE MORE FLEXIBLE IN MANAGING AND ACHIEVING PROGRAM GOALS.

- HISTORICALLY, THE DEPARTMENT OF DEFENSE (DOD) AND THE U.S. GOVERNMENT HAS BEEN THE LARGEST PURCHASER OF SOFTWARE THROUGH THE 1990'S. MOST GOVERNMENT CONTRACTS FOLLOWED THE WATERFALL LIFECYCLE MODEL OR A DERIVATIVE OF BASED ON THE REQUIREMENTS SPECIFIED IN THE REQUEST FOR PROPOSAL (RFP) THAT A CONTRACTOR RESPONDS TOO. UNFORTUNATELY, MANY OF THE PROGRAMS DURING THE 1970'S AND MID 1980'S OVER RAN DUE TO COST AND SCHEDULE ISSUES AND SOME FAILED TO PRODUCE AN END PRODUCT.

- TODAY THE DOD IS MOVING MORE TOWARDS A DEVOPS ENVIRONMENT IN WHICH THEY MANAGE THE PROJECT INTERNALLY AND HIGHER CONTRACTORS TO PERFORM THE WORK UNDER THEIR DIRECTION WITH ASSISTANCE FROM TECHNICAL EXPORTS FROM THE CUSTOMER. THIS IS ENABLING THE DOD TO MOVE MORE INTO THE AGILE DEVELOPMENT ARENA.

REFERENCES

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

CMMI PROCESS MODEL - BY SARAH K. WHITE, SENIOR WRITER, CIO, MAR. 16, 2018

12207-2017 ISO/IEC/IEEE SYSTEMS AND SW ENGINEERING STANDARD

SOFTWARE ENGINEERING, A PRACTITIONER’S APPROACH, EIGHTH EDITION, BY ROGER S. PRESSMAN AND BRUCE R. MAXIM, COPYRIGHT 2015 BY MCGRAW HILL, NEW YORK, NY