

PROCESS

AGILE & ITERATIVE DEVELOPMENT (CHAPTER 1)

A MANAGER'S GUIDE BY: CRAIG LARMAN

WEEK 2: INTRODUCTION – THE KEY TO SUCCESSFUL PROGRAM
MANAGEMENT

BY: JOSEPH MARTINAZZI

LEAD WITH HONESTY AND INTEGRITY

PROGRAM MANAGEMENT IS RESPONSIBLE FOR DEFINING **CLEAR OBJECTIVES** FOR THE PROGRAM TO ENSURE THE END-PRODUCT WILL MEET OR EXCEED THE CUSTOMER'S EXPECTATION (RELIABILITY, SECURITY, AND SAFETY).

YOUR JOB AS A SOFTWARE MANAGER IS TO NOT ONLY UNDERSTAND PROGRAM OBJECTIVES, BUT TO ALWAYS BE **LOOKING AHEAD** IN YOUR PLANNING SO THAT YOUR TEAM CAN BE **ADAPTIVE** AND NOT **REACTIVE** WHEN ISSUES ARISE.

TO BE A SUCCESSFUL LEADER YOU NEED TO:

- EFFECTIVELY COMMUNICATE TO BOTH MANAGEMENT AND YOUR TEAM,
- DON'T CREATE A PLAN THAT REQUIRES YOUR TEAM TO **ALWAYS** WORK AT 110% (*SAVE IT FOR WHEN MURPHY'S LAW HITS*),
- UTILIZE YOUR STAFF TO THE BEST OF THEIR ABILITIES (*EVERYONE CAN BE A VALUABLE CONTRIBUTOR*),
- CREATE A TEAM ENVIRONMENT (*YOU SUCCEED OR FAIL AS A TEAM*),
- LEAD BY EXAMPLE (*DON'T ASK YOUR TEAM TO DO SOMETHING YOU WOULDN'T DO YOURSELF*),
- BECOME **CAPTAIN** AMERICA - **SHIELD** YOUR TEAM FROM PROGRAM POLITICS
- HAVE AN OUTWARDLY POSITIVE ATTITUDE!

EVERYTHING IS AWESOME, EVERYTHING IS COOL WHEN YOUR PART OF JOE'S TEAM. EVERYTHING IS AWESOME WHEN YOUR LIVING THE PROGRAM X-Y-Z DREAM !

WHAT DO WE HAVE TO DO - KEEP MOVING FORWARD!

INITIAL PROGRAM PLANNING

As part of the initial program planning effort, key processes must be established that take the following into account:

- The contractual requirements the system under development must meet.
- A program “Change Management Plan” that describes how program artifacts will be controlled and how change will be managed during the development of the product (in-house) as well as for the delivered product (customer facility) if necessary.
- The overall program testing strategy based on the contract and how the end-products will be sold off to the customer.
- The time period (**schedule**) in which the system must be completed including any interim milestones that must be met.
- The **budget** allocated to the effort you will be responsible for managing. *This includes task your team is directly responsible for as well as support tasks for other organizations.*
- The facilities (lab environment) you will need to support the development of the end-product.
- A staffing plan to efficiently staff the program to meet cost and schedule requirements.
- Risks associated with achieving program goals for the tasks your team has been allocated.

INTEGRATED PROGRAM MANAGEMENT

PROGRAM ORGANIZATION

KEY PROGRAM PROCESSES:

1. PROGRAM REQUIREMENTS DOCUMENT,
2. PROGRAM CHANGE MANAGEMENT PLAN,
3. PROGRAM VERIFICATION & VALIDATION PLAN,
4. **PROGRAM SCHEDULE AND MILESTONES (IMP AND IMS),**
5. PROGRAM WORK BREAKDOWN STRUCTURE/COST COLLECTION METHOD,
6. PROGRAM STAKE HOLDER INVOLVEMENT PLAN,
7. PROGRAM QUALITY ASSURANCE PLAN, AND
8. PROGRAM RISK MANAGEMENT PLAN.

INTEGRATED MASTER PLAN & MASTER SCHEDULE

A **PROGRAM SCHEDULE** IS TYPICALLY REFERRED TO AS AN **INTEGRATED MASTER SCHEDULE (IMS)** AND IS BASED OFF A PROGRAM **INTEGRATED MASTER PLAN (IMP)** THAT WAS CREATED AT THE TIME OF THE PROPOSAL.

ACCORDING TO THE “MITRE SYSTEMS ENGINEERING GUIDE”, A DEFINITION FOR AN IMP AND AN IMS ARE AS FOLLOWS:

- “THE **IMP** COMPRISES A HIERARCHY OF PROGRAM EVENTS, IN WHICH EACH EVENT IS SUPPORTED BY SPECIFIC ACCOMPLISHMENTS, AND EACH ACCOMPLISHMENT IS BASED ON SATISFYING SPECIFIC CRITERIA TO BE CONSIDERED COMPLETE. THE IMP IS AN EVENT-DRIVEN PLAN IN WHICH THE EVENTS ARE NOT TIED TO CALENDAR DATES; THEY ARE TIED TO THE ACCOMPLISHMENT OF A TASK OR WORK PACKAGE AS EVIDENCED BY THE SATISFACTION OF THE SPECIFIED CRITERIA FOR THAT ACCOMPLISHMENT.”
- “THE **IMS** IS AN INTEGRATED, NETWORKED SCHEDULE OF ALL THE DETAILED, DISCRETE WORK PACKAGES AND PLANNING PACKAGES (OR LOWER LEVEL TASKS OF ACTIVITIES) NECESSARY TO SUPPORT THE IMP'S EVENTS, ACCOMPLISHMENTS, AND CRITERIA. THE IMS IS DEVELOPED FROM THE IMP, MAJOR CONTRACTOR EVENTS, ACCOMPLISHMENTS, ENTRANCE CRITERIA, EXIT CRITERIA, AND THE WBS, WHICH DEFINES THE PROGRAM WORK STRUCTURE AND WORK PACKAGES. THE IMS IS TIME DRIVEN, TIED TO CALENDAR DATES, AND SHOULD BE DEFINED TO THE LEVEL OF DETAIL NECESSARY FOR PROGRAM EXECUTION.”

REFERENCES AND RESOURCES

DAU, DEFENSE ACQUISITION GUIDEBOOK, CHAPTER 11.3.1.4.2, INTEGRATED MASTER SCHEDULE (IMS), ACCESSED JUNE 7, 2016.

DEPARTMENT OF DEFENSE, OCTOBER 21, 2005, **INTEGRATED MASTER PLAN AND INTEGRATED MASTER SCHEDULE: PREPARATION AND USE GUIDE**, VER. 0.9, ACCESSED SEPTEMBER 14, 2017.

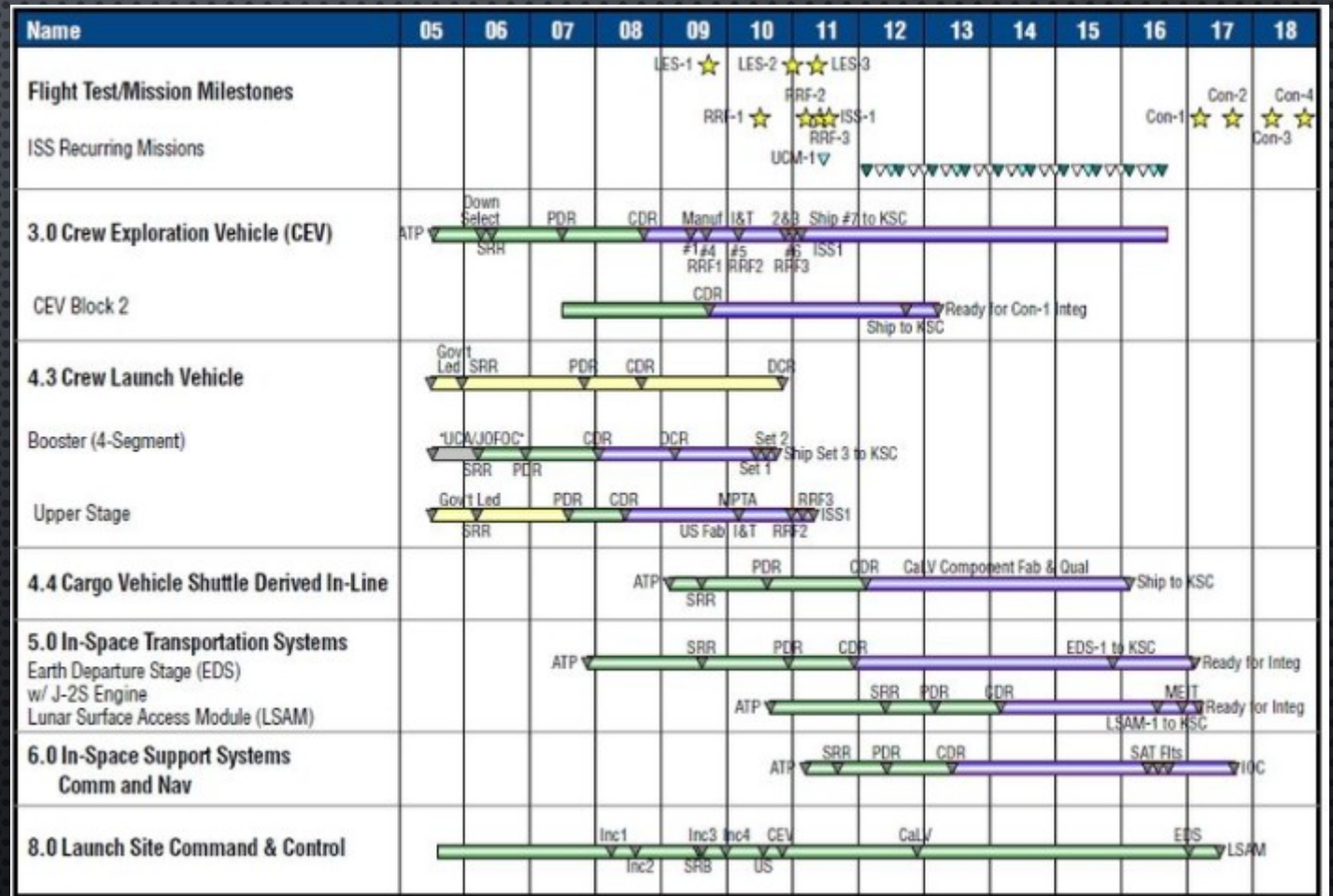
INCOSE, 2015, SYSTEMS ENGINEERING HANDBOOK, A GUIDE FOR SYSTEM LIFE-CYCLE PROCESSES AND ACTIVITIES, FOURTH ED., INCOSE-TP-2003-002-04.

PROJECT MANAGEMENT INSTITUTE (PMI), 2013, **STANDARD FOR PROGRAM MANAGEMENT**, THIRD ED

INTEGRATED MASTER PLAN & MASTER SCHEDULE

EXAMPLE IMP AND IMS

The high-level program schedule typically comes directly from the contract and is based on the proposed schedule in the Request for Proposal (RFP) that was refined in your company's proposal.



INTEGRATED MASTER PLAN & MASTER SCHEDULE

THE IMS IS ONE OF THE MOST IMPORTANT PROGRAM MANAGEMENT TOOLS TO EFFECTIVELY MANAGING THE PROGRAM.

- IT IS USED TO REPORT PROGRESS TO THE CUSTOMER, AND
- IT IS USED BY PROGRAM MANAGEMENT TO DETERMINE IF THE PROGRAM IS ON SCHEDULE.

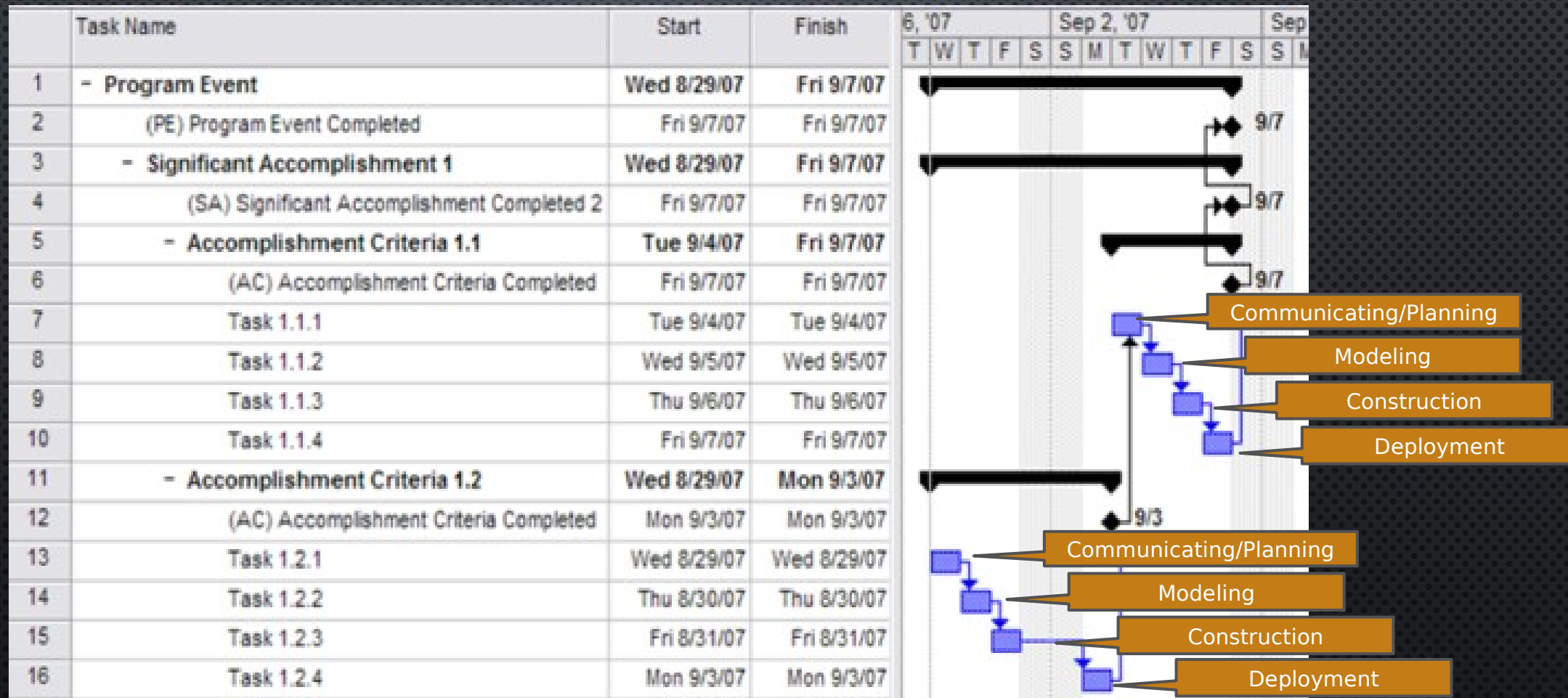
AT THE END OF STARTUP PLANNING THE IMS WILL BE BASELINED WITH ALL PROGRAM TASKS IDENTIFIED IN EITHER A

- **WORK PACKAGE** (IMPLYING THE TASK CAN BE WORKED WITHIN THE NEXT SEVERAL MONTHS) OR IN A
- **PLANNING PACKAGE** (TASKS TO BE WORKED AT SOME FUTURE DATE ON THE PROGRAM).
- SOME WORK MAY ALSO BE PLANNED AS A LEVEL OF EFFORT (LOE).
MY RECOMMENDATION IS TO AVOID USING LOE.

EACH TASK WILL ALSO HAVE

- AN ASSOCIATED DURATION (HOURS),
- AN ASSOCIATED BUDGET (DOLLARS),
- AND A MEASURE OF HOW STATUS WILL BE REPORTED (E.G. 50-50, PERCENT COMPLETE, ETC.).
MY RECOMMENDATION IS TO ALWAYS DEFAULT TO USING THE PERCENT COMPLETE METHOD AS THE PREFERRED METHOD TO MEASURE PROGRESS ON A TASK.

INTEGRATED MASTER PLAN & MASTER SCHEDULE



INTEGRATED MASTER PLAN & MASTER SCHEDULE

PROGRAM STATUS CAN BE COLLECTED FROM AN EARNED VALUE MANAGEMENT SYSTEM AND PULLED INTO THE IMS ON A WEEKLY BASIS TO DETERMINE THE PROGRESS OF ALL TASKS CURRENTLY IN WORK.

- TASKS WILL EITHER SHOW THEY ARE **AHEAD** OF SCHEDULE, **ON** SCHEDULE, OR **BEHIND** SCHEDULE.
- ON A ROLL-UP LEVEL, THIS WILL SHOW IF THE OVERALL PROGRAM IS AHEAD, ON, OR BEHIND SCHEDULE.
IF THE PROGRAM SCHEDULE IS DELAYED; COSTS BECOME UNCONTROLLABLE.

REMEMBER **PROGRAM BUDGETS ARE CALENDARIZED** BASED ON THE IMS. AS A RESULT, IF SEVERAL OF THE TASKS YOU ARE RESPONSIBLE FOR SLIP INTO A DIFFERENT CALENDAR YEAR, THEY MAY OVER-RUN BASED ON DOLLARS EVEN IF THEY COMPLETE WITHIN THE HOURS ALLOCATED TO COMPLETE THE WORK ASSOCIATED WITH THE TASK.

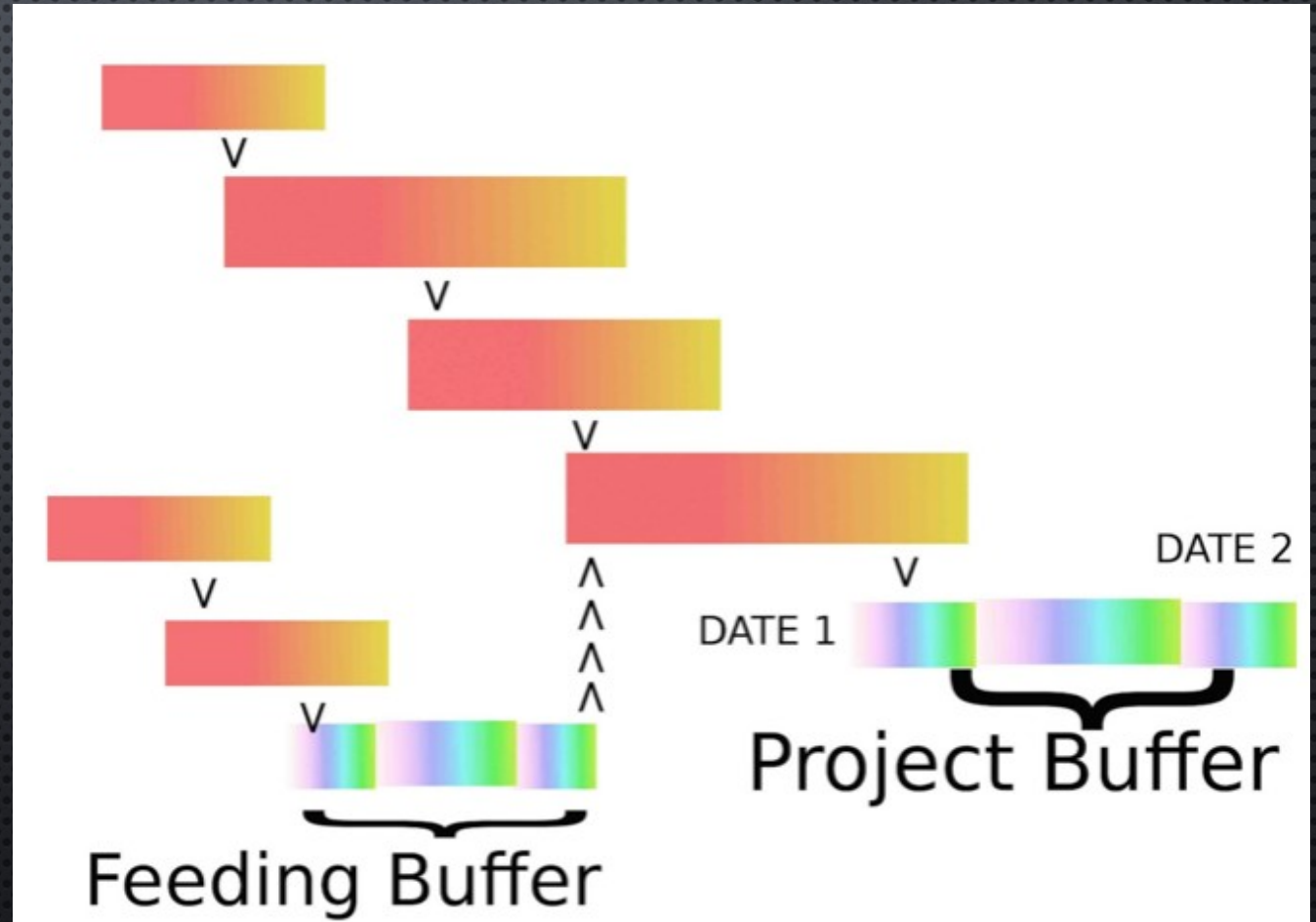
THE ABILITY TO MONITOR ALL TASKS WITHIN A SCHEDULE GIVES PROGRAM MANAGEMENT THE ABILITY TO DETERMINE **CRITICAL PATHS THAT DRIVE THE PROGRAM**. TYPICALLY, PROGRAMS WILL MONITOR THE TOP THREE CRITICAL PATHS THROUGH THE PROGRAM. TYPICALLY SOFTWARE DEVELOPMENT, INTRA-SITE COMMUNICATION DEVELOPMENT, OR HARDWARE PROCUREMENT WILL DRIVE THE PROGRAM CRITICAL PATHS.

IF A TASK YOU ARE RESPONSIBLE FOR IS ON ONE OF THE PROGRAM CRITICAL PATHS YOU WILL NEED TO BE PREPARED TO ALWAYS HAVE A SOLUTION AS TO HOW TO MINIMIZE THE SCHEDULE TO COMPLETE THAT TASK.

MANAGING PROGRAM SCHEDULE BUFFERS

AN EFFECTIVE WAY TO MINIMIZE PROGRAM SCHEDULE IMPACT AND KEEP INSIGHT INTO TASK IMPLEMENTATION IS BY UTILIZING A PROCESS KNOWN AS "CRITICAL CHAIN PROJECT MANAGEMENT" WHICH FOCUSES ON MANAGING PROGRAM BUFFERS AND RESOURCES (STAFFING AND LAB EQUIPMENT).

PROGRAMS THAT SUCCESSFULLY MANAGE THEIR BUFFERS WILL END UP BEING SUCCESSFUL.



MANAGING PROGRAM SCHEDULE BUFFERS

SW DEVELOPERS ARE **ALWAYS OPTIMISTIC** WHEN ASKED HOW LONG IT WILL TAKE TO COMPLETE A TASK. THEIR ANSWER NEVER ACCOUNTS FOR MURPHY'S LAW (IF SOMETHING CAN GO WRONG IT WILL GO WRONG).

AS A MANAGER YOU NEED TO ACCOUNT FOR MURPHY'S LAW BY MANAGING THE DEVELOPMENT SCHEDULE. ONE OF THE MOST EFFECTIVE WAYS TO DO THIS IS TO CREATE A BUILD PLAN WITH A SCHEDULE BUFFER BUILT IN.

FOR EXAMPLE, IF YOU SAY BUILD 1 WILL TAKE 3 MONTHS TO COMPLETE IN THE BUILD PLAN.

- WHEN YOU ACTUALLY PLAN THE ACTIVITY IN THE IMS YOU WILL NEED TO BUILD IN A 3 WEEK BUFFER BY ACTUALLY PLANNING ALL OF THE TASKS TO COMPLETE 3 WEEKS AHEAD OF THE ADVERTISED FINISH OF THE BUILD BY USING SCHEDULE DEPENDENCIES (BASED ON RESOURCE AVAILABILITY) VS. TECHNICAL DEPENDENCIES (BASE ON COMPONENT DESIGN DEPENDENCIES).
- ANOTHER WAY TO ACHIEVE THIS IS TO HAVE TASKS TO FIX ISSUES ASSOCIATED WITH EACH BUILD THAT THE DEVELOPMENT WORK FEEDS INTO.

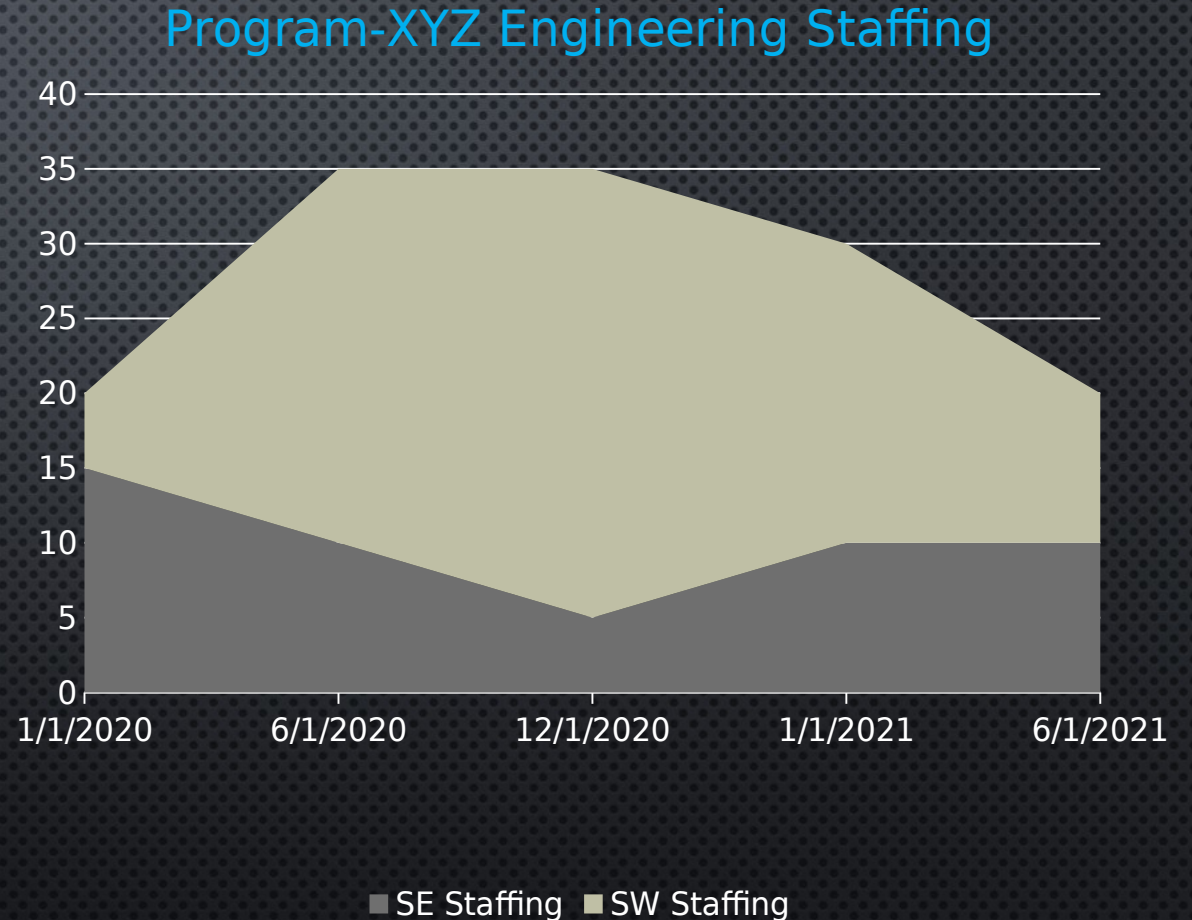
PROGRAM STAFFING NEEDS

THE IMS IS ALSO A KEY TOOL TO IDENTIFY IF PROGRAM STAFFING WILL IMPACT PRODUCT DEVELOPMENT FROM BOTH A SCHEDULE AND COST PERSPECTIVE.

SINCE THE IMS PROVIDES A CALENDARIZED ROLE UP AS TO WHEN TASKS NEED TO BE COMPLETED (HOURS) A STAFFING PROFILE CAN BE EXTRACTED FROM THE IMS WHICH INDICATES HOW MANY PEOPLE WILL BE NEEDED TO COMPLETE THE WORK AND THE DURATION THOSE INDIVIDUALS WILL BE NEEDED FOR.

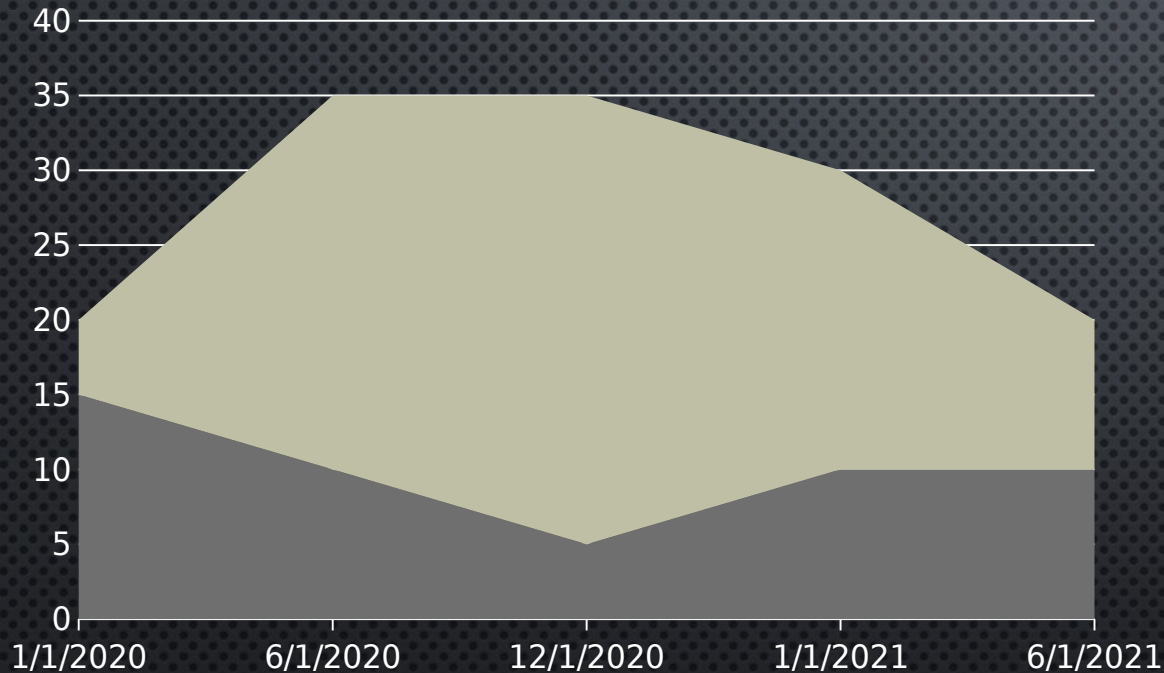
THIS STAFFING PROFILE WILL INDICATE HOW MANY PEOPLE YOU NEED TO COMPLETE THE TASKS WITHIN YOUR SCHEDULE.

- IF YOUR ORGANIZATION IS OVER STAFFED, YOU SHOULD BE COMPLETING TASKS AHEAD OF SCHEDULE TO STAY ON BUDGET.
- IF YOUR ORGANIZATION IS CORRECTLY STAFFED, YOU SHOULD BE COMPLETING TASK ON OR AHEAD OF SCHEDULE (ASSUMING THE TASKS ARE CORRECTLY SCOPED),
- IF YOUR ORGANIZATION IS UNDERSTAFFED YOU COULD END UP IN A SITUATION IN WHICH THE ENTIRE PROGRAM SCHEDULE FALLS BEHIND IF YOU CAN NOT COMPLETE THE WORK AS PLANNED.



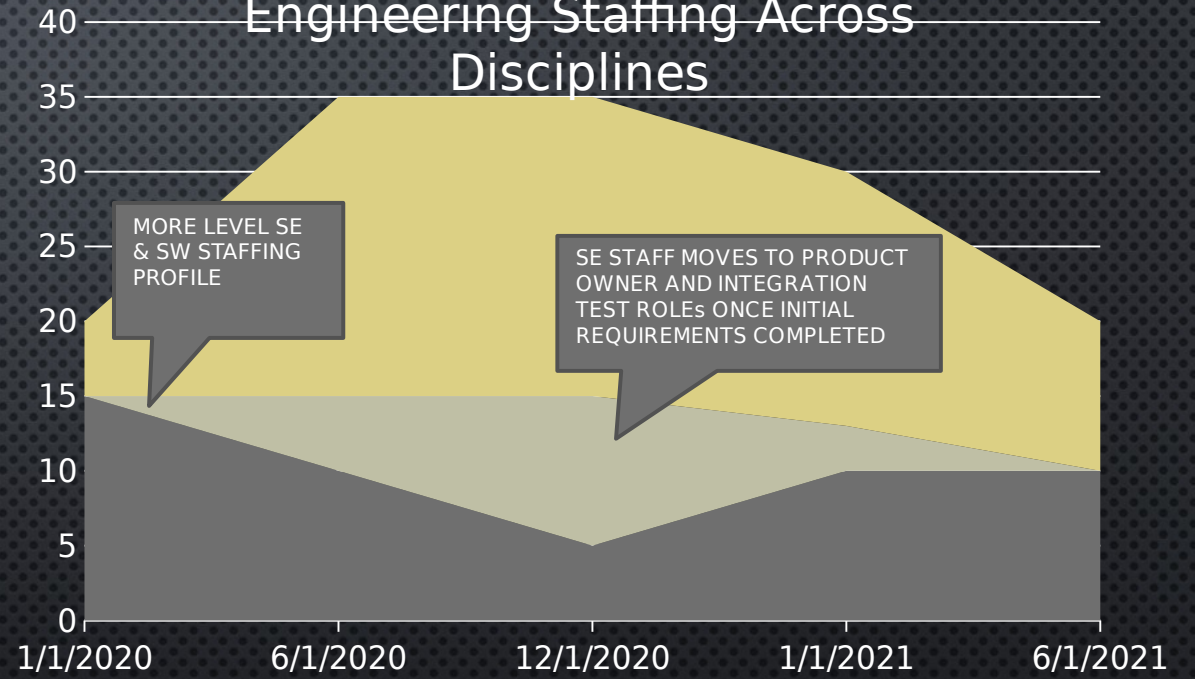
PROGRAM STAFFING NEEDS

Program-XYZ Engineering Staffing



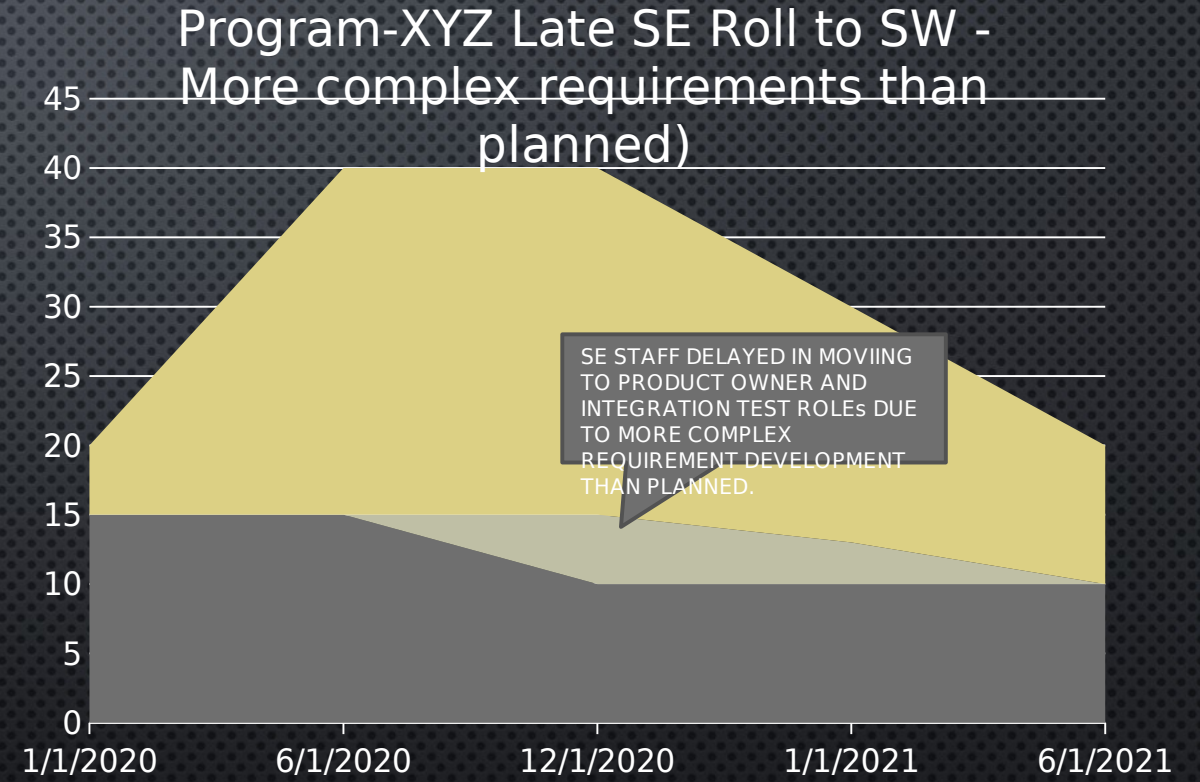
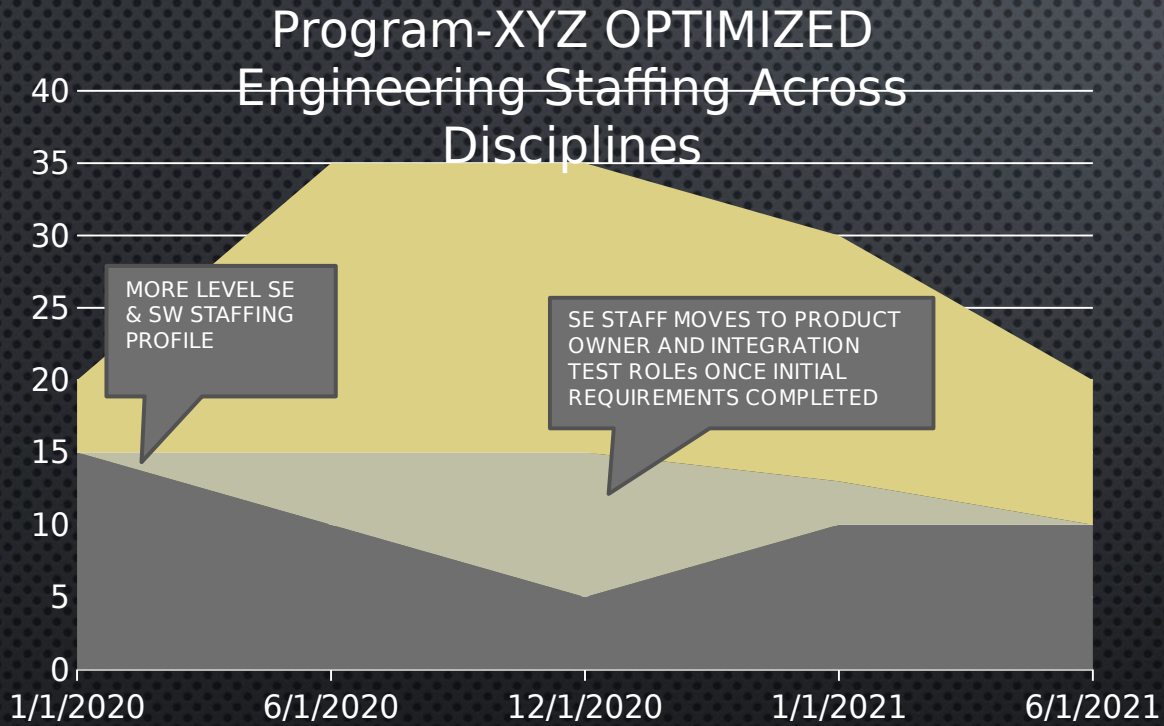
■ SE Staffing ■ SW Staffing

Program-XYZ OPTIMIZED Engineering Staffing Across Disciplines



■ SE Staffing ■ SE-SW Staffing ■ SW Staffing2

PROGRAM STAFFING NEEDS



■ SE Staffing ■ SE-SW Staffing ■ SW Staffing2

■ SE Staffing ■ SE-SW Staffing ■ SW Staffing2

SW PROJECT PLANNING - OVERVIEW

THE INITIAL PLANNING STAGE OF A PROGRAM HAPPENS WITHIN A VERY SHORT PERIOD OF TIME.

THE **KEY OBJECTIVES** FOR THE SOFTWARE PROGRAM MANAGER (SPM) DURING THIS PHASE OF THE PROGRAM INCLUDES:

- DETAIL PLANNING TASKS THAT NEED TO BE COMPLETED WITHIN THE PROGRAM SCHEDULE.
- ACCOUNTING FOR THE ASSOCIATED RISKS IN THE DEVELOPMENT OF THOSE TASKS,
- DETERMINING IF THOSE RISKS NEED TO BE REALIZED OR MITIGATED AS PART OF THE INITIAL PLAN,
- SECURING THE NECESSARY BUDGET TO COMPLETE ALL TASKS ASSIGNED TO THE ORGANIZATION,
- SECURING THE SCHEDULE NEEDED TO DEVELOP THE PRODUCT, AND
- CREATING ALL STARTUP PROCESSES NEEDED TO EXECUTE THE SOFTWARE DEVELOPMENT EFFORT
- SPENDING AS LITTLE MONEY AS POSSIBLE DURING THIS PHASE OF THE PROGRAM.

SW PROJECT PLANNING - ASSUMPTIONS

FOR THE PURPOSE OF THIS DISCUSSION, WE ARE GOING TO ASSUME

- THE COMPANY YOU WORK FOR HAS AN EXISTING PRODUCT THEY PLAN ON LEVERAGING TO MEET MOST OF THE REQUIREMENTS SPECIFIED IN THE CUSTOMER'S RFP.
- THIS PRODUCT HAS AN ASSOCIATED SOFTWARE BASELINE THAT CAN BE LEVERAGED AS A STARTING POINT FOR THE PROGRAM.
- SINCE YOUR COMPANY HAS A CMMI LEVEL 5 RATING, WE KNOW THAT THE COMPANY HAS ESTABLISHED METRICS ON BOTH SOFTWARE SIZE AND SOFTWARE PRODUCTIVITY THAT WENT INTO THE PROPOSAL ESTIMATE (MEANING THE BUDGET YOUR ORGANIZATION BID SHOULD BE SUFFICIENT TO COVER THE COST OF IMPLEMENTING THE SYSTEM).

SW PROJECT PLANNING – INITIAL STAFFING

THE SOFTWARE STAFF DURING THIS STAGE OF THE PROGRAM NEEDS TO BE MINIMALIZED TO PREVENT EROSION OF BUDGET DURING THE PLANNING STAGE.

MY RECOMMENDATION IS THAT IT INCLUDES THE SPM, A SOFTWARE TECHNICAL LEAD (STD) AND ADDITIONAL STAFF ONLY AS NEEDED TO SUPPORT THE PROGRAM STARTUP ACTIVITIES.

REMEMBER THE JOB OF THE SPM IS TO ALWAYS LOOK AHEAD AND MITIGATE PROGRAM RISK.

- YOUR SUCCESS WILL ULTIMATELY BE DETERMINED BASED ON YOUR ABILITY TO COMPLETE THE PROGRAM WITHIN COST AND SCHEDULE.
- THE ONLY ONE THAT WILL REMEMBER CHALLENGES YOU HAVE TO OVERCOME IS YOURSELF!
- CHALLENGES MAY COME FROM THE PROGRAM OFFICE (PMO) OR FROM YOU OWN ORGANIZATION WITHIN THE COMPANY. INITIAL PROGRAM STAFFING MAY BE THE FIRST CHALLENGE YOU HAVE TO OVERCOME.
 - IF YOUR ORGANIZATION FORCES YOU TO STAFF UP EARLIER THAN YOU WANT, MAKE SURE YOU HAVE A WAY TO ACCURATELY COLLECT COSTS, HAVE TASKS INDIVIDUALS CAN WORK TO ACHIEVE PROGRAM GOALS, AND HAS MANAGEMENT OVERSITE.
 - IF YOUR ORGANIZATION DOESN'T HAVE THE STAFF TO SUPPORT YOUR PROGRAM, BRING SOMEONE ON BOARD WHO IS CAPABLE OF OPENING JOB REQUISITIONS INTERVIEWING AND GETTING THE STAFFING TALENT YOUR PROGRAM WILL NEED.

SW PROJECT PLANNING – ESTABLISHING A BASELINE SCHEDULE AND BUDGET

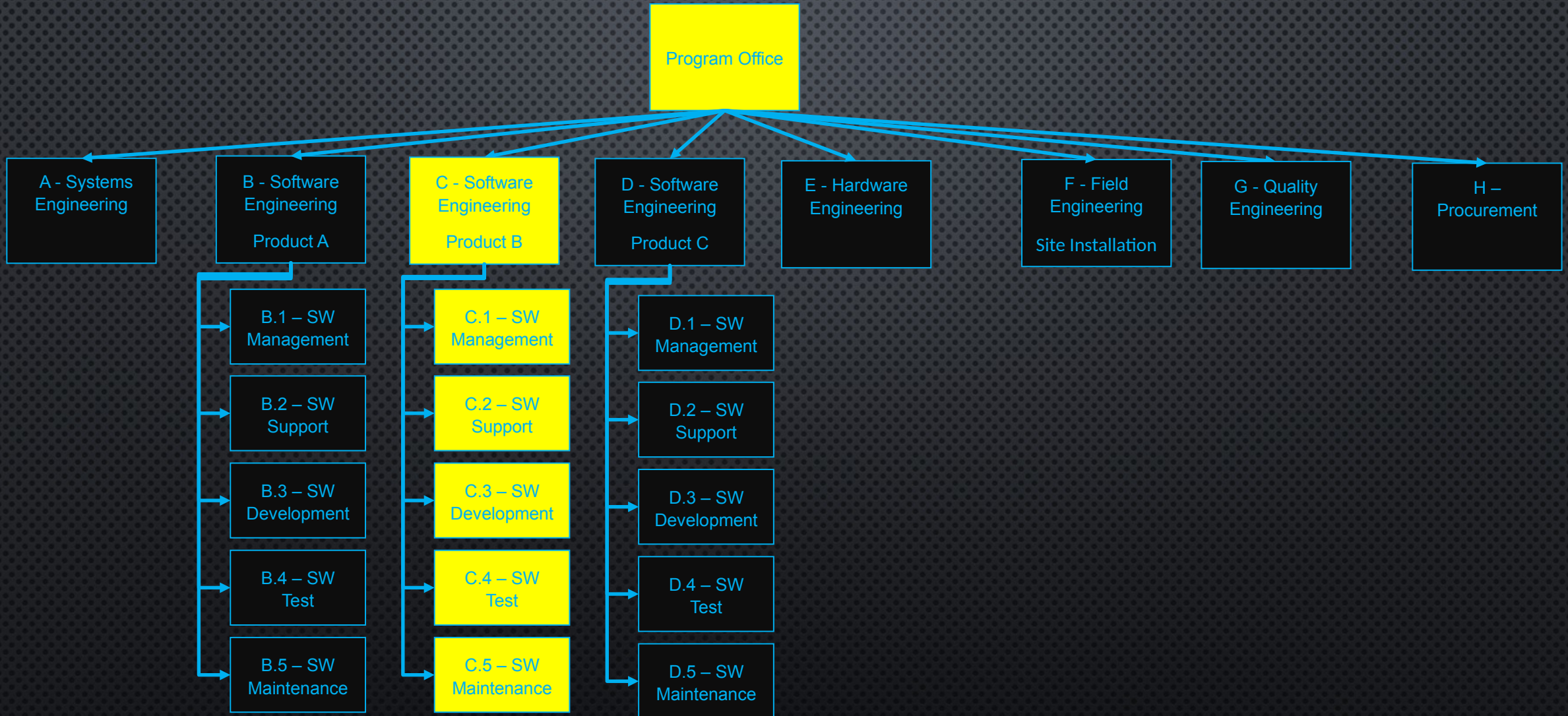
YOUR GOAL AS A SOFTWARE MANAGER WILL BE TO **UNDERSTAND THE IMP AND IMS** AND **CREATE A SOFTWARE BUILD PLAN** (SBP) THAT CAN FIT INTO THE PROGRAM SCHEDULE THAT ACCOUNTS FOR ALL THE TASKS YOUR TEAM WILL BE RESPONSIBLE TO COMPLETE OR SUPPORT.

IN ADDITION, YOU WILL BE RESPONSIBLE FOR UNDERSTANDING THE SOFTWARE BID YOUR ORGANIZATION PUT FORTH IN RESPONSE TO THE CUSTOMER'S RFP AND JUSTIFY THE COST TO COMPLETE EACH ACTIVITY TO THE PROGRAM OFFICE PRIOR TO THEM ALLOCATING YOUR BUDGET.

ALL ACTIVITIES WITHIN THE PROPOSAL ARE ORGANIZED IN A WORK BREAKDOWN STRUCTURE (WBS) THAT MAPS HOURS AND BUDGETS TO ACTIVITIES THAT NEED TO BE PERFORMED AS PART OF PROGRAM EXECUTION. THESE ACTIVITIES INCLUDE:

- SOFTWARE MANAGEMENT ACTIVITIES,
- SOFTWARE SUPPORT ACTIVITIES,
- SOFTWARE DEVELOPMENT ACTIVITIES,
- SOFTWARE TEST ACTIVITIES, AND
- SOFTWARE MAINTENANCE ACTIVITIES

SW PROJECT PLANNING – EXAMPLE PROGRAM WBS



SW PROJECT PLANNING – ESTABLISHING A BASELINE SCHEDULE AND BUDGET

IF YOU'RE LUCKY, AN ARTIFACT THAT MAY HAVE BEEN CREATED DURING THE PROPOSAL PHASE IS A THIN SPECIFICATION (SPEC.).

THIS DOCUMENT TYPICALLY GOES ALONG WITH A BID AN ORGANIZATION PUT FORTH THAT ESTIMATED HOW MUCH EFFORT IT WOULD TAKE TO IMPLEMENT THE CAPABILITIES/FEATURES OF THE END-PRODUCT SPECIFIED IN THE RFP.

DURING STARTUP, THE PROPOSAL INFORMATION IS USED TO:

1. CREATE A SOFTWARE BUILD PLAN (SBP),
2. SUPPORT DETAILED IMS PLANNING,
3. JUSTIFY YOUR ORGANIZATIONS BUDGET TO PROGRAM MANAGEMENT,
4. SOLIDIFY YOUR THE WBS STRUCTURE YOU WANT TO USE TO MANAGE THE PROGRAM, AND
5. CREATE A SOFTWARE DEVELOPMENT PLAN (SDP).

SW PROJECT PLANNING – CREATING THE SBP

1. CREATE A SOFTWARE BUILD PLAN (SBP)

THE SPM NEEDS TO WORK WITH THE STD IN LAYING OUT THE PRELIMINARY SBP WHICH SIMPLY BREAKS THE SOFTWARE INTO MAJOR CAPABILITIES OR FEATURES AND IDENTIFIES THE SEQUENCE IN WHICH THOSE CAPABILITIES/FEATURES WILL BE DEVELOPED (FROM AN AGILE PERSPECTIVE YOU CAN THINK OF THIS EXERCISE AS CREATING A PRODUCT BACKLOG AND ESTIMATING THE EFFORT ASSOCIATED WITH DEVELOPING THESE HIGH-LEVEL FEATURES)

- 1) THE FIRST STEP IN CREATING A SOFTWARE BUILD PLAN IS FOR THE SPM, STD, AND OTHER TECHNICAL LEADS WITHIN THE ORGANIZATION TO CREATE A **BASIS OF ESTIMATE (BOE)** FOR EACH CAPABILITY/FEATURE BEING DEVELOPED BASED ON ESTABLISHED SIZE AND PRODUCTIVITY METRICS (HOW MANY LINES OF CODE WILL BE DEVELOPED AND MANY HOURS IT WILL TAKE TO COMPLETE THE TASK).
- 2) EACH BOE SHOULD ALSO HAVE A HIGH-LEVEL DESCRIPTION OF WHAT EACH CAPABILITY/FEATURE WILL ACHIEVE.

SW PROJECT PLANNING – CREATING THE SBP

3) SOFTWARE BUILD PLAN (SBP) STRUCTURE

- POINT OF DEPARTURE BUILD – TYPICALLY CONSIST OF CREATING A SOFTWARE BASELINE TO SUPPORT THIS PROGRAM FROM A PRODUCT LINE BASELINE. TYPICAL ACTIVITIES WOULD INCLUDE PORTING THE SOFTWARE AND ASSOCIATED TEST BED AND CREATING ADAPTATION DATA TO SUPPORT THE SPECIFIC PROGRAM.
- FRAMEWORK ENHANCEMENT BUILD – TYPICALLY CONSIST OF MODIFICATIONS TO THE PRODUCT LINE ARCHITECTURE TO SUPPORT THE NEW COMPONENTS OR FEATURES (EXAMPLES INCLUDE CORE FRAMEWORK ENHANCEMENTS, MESSAGING FRAMEWORK ENHANCEMENTS, AND PERFORMANCE ENHANCEMENTS).
- CAPABILITY/FEATURE BUILDS – ONE OR MORE BUILDS THAT ADD CAPABILITIES/FEATURES BASED ON THE CONCEPT OF MUST-HAVE FEATURES, NICE-TO-HAVE FEATURES, AND WOULD-BE-GREAT-TO-HAVE FEATURES IF POSSIBLE.

SW PROJECT PLANNING – CREATING THE SBP

4) SOFTWARE BUILD PLAN (SPB) CONTENT

BUILD 0 – POINT OF DEPARTURE BUILD

USE CASE 0.1 – DESCRIPTION OF CAPABILITY/FEATURE

USE CASE 0.2 – DESCRIPTION OF CAPABILITY/FEATURE

USE CASE 0.3 – DESCRIPTION OF CAPABILITY/FEATURE

BUILD 2 – CAPABILITY/FEATURE A BUILD

USE CASE 2.1 – DESCRIPTION OF CAPABILITY/FEATURE

USE CASE 2.2 – DESCRIPTION OF CAPABILITY/FEATURE

USE CASE 2.3 – DESCRIPTION OF CAPABILITY/FEATURE

BUILD 1 – FRAMEWORK ENHANCEMENT BUILD

USE CASE 1.1 – DESCRIPTION OF CAPABILITY/FEATURE

USE CASE 1.2 – DESCRIPTION OF CAPABILITY/FEATURE

USE CASE 1.3 – DESCRIPTION OF CAPABILITY/FEATURE

BUILD 3 – CAPABILITY/FEATURE B BUILD

USE CASE 3.1 – DESCRIPTION OF CAPABILITY/FEATURE

USE CASE 3.2 – DESCRIPTION OF CAPABILITY/FEATURE

USE CASE 3.3 – DESCRIPTION OF CAPABILITY/FEATURE

BUILD 4 – CAPABILITY/FEATURE C BUILD

USE CASE 4.1 – DESCRIPTION OF CAPABILITY/FEATURE

USE CASE 4.2 – DESCRIPTION OF CAPABILITY/FEATURE

USE CASE 4.3 – DESCRIPTION OF CAPABILITY/FEATURE

SW PROJECT PLANNING – DETAILED PLANNING

2. SUPPORT DETAILED IMS PLANNING

ONCE YOUR SBP IS COMPLETED, YOU CAN BEGIN CREATING TECHNICAL AND SCHEDULE DEPENDENCIES BETWEEN THE DEVELOPMENTAL ACTIVITIES IN THE IMS.

1) **AS PART OF THE INITIAL PLANNING STAGE** – ALL SOFTWARE ACTIVITIES MUST BE MAPPED TO A TASK WITHIN THE IMS AND HAVE HOURS AND BUDGET ASSOCIATED WITH THE TASK. ACTIVITIES WITHIN THE IMS ARE MODELED AS A

- WORK PACKAGE – MEANING THE WORK DEFINED WITHIN THIS WORK PACKAGE IS AUTHORIZED TO BE WORKED ON TODAY OR WITHIN THE NEXT FEW MONTHS.
- PLANNING PACKING – MEANING THE WORK DEFINED WITHIN THIS WORK PACKAGE WILL BE AUTHORIZED TO BE PERFORMED SOMETIME IN THE FUTURE.
- LEVEL-OF-EFFORT – MEANING CREDIT FOR THE WORK AUTOMATICALLY OCCURS WHETHER OR NOT ANYTHING GETS ACCOMPLISHED OR AN INDIVIDUAL WORKS A TASK (E.G. SPM OR SUPPORT LABOR)

SW PROJECT PLANNING – DETAILED PLANNING

2) MODELING EACH CATEGORY WITHIN THE SOFTWARE BID IN THE IMS

- SOFTWARE MANAGEMENT AND SUPPORT ACTIVITIES – ARE PLANNED IN THE IMS AS EITHER A WORK/PLANNING PACKAGE TASK OR A LOE TASK AND LINKED SEQUENTIALLY OVER A PERIOD OF PERFORMANCE.
- SOFTWARE DEVELOPMENT ACTIVITIES – ARE PLANNED IN THE IMS AS EITHER A WORK/PLANNING PACKAGE DISCRETE TASK AND LINKED BASED ON THEIR TECHNICAL AND SCHEDULE DEPENDENCIES TO OTHER DEVELOPMENTAL DISCRETE TASKS.

NOTE: IMS LOGIC MUST TAKE THE SOFTWARE DEVELOPMENT MODEL SPECIFIED WITHIN THE SOFTWARE DEVELOPMENT PLAN (SDP) INTO ACCOUNT. FOR EXAMPLE, IF THE SDP STATES THAT AN ITERATIVE APPROACH FOR SW DEVELOPMENT IS BEING USED THEN THE IMS LOGIC MUST REFLECT THAT MODEL AS PART OF THE DEVELOPMENT SCHEDULE.

- SOFTWARE TEST ACTIVITIES – ARE PLANNED IN THE IMS AS EITHER A WORK/PLANNING PACKAGE DISCRETE TASK AND LINKED BASED ON THEIR TECHNICAL AND SCHEDULE DEPENDENCIES TO DEVELOPMENTAL DISCRETE TASKS.
- SOFTWARE MAINTENANCE ACTIVITIES – ARE PLANNED IN THE IMS AS EITHER A WORK/PLANNING PACKAGE DISCRETE TASK AND LINKED BASED ON THEIR TECHNICAL AND SCHEDULE DEPENDENCIES TO DEVELOPMENTAL AND TEST DISCRETE TASKS.

THE SUM OF ALL TASKS IN THE BASELINED IMS WILL TRANSLATE TO THE BASELINE SCHEDULE (IN HOURS) AND BUDGET (IN DOLLARS) TO COMPLETE THE WORK.

MY RECOMMENDATION IS TO NEVER USE AN LOE TASK IN THE IMS UNLESS YOU ARE FORCED TO DO SO!

SW PROJECT PLANNING – ESTABLISHING YOUR COST BASELINE

3. JUSTIFY YOUR ORGANIZATIONS BUDGET TO PROGRAM MANAGEMENT.

1) EACH TASK WILL NEED TO HAVE A **BASIS OF ESTIMATE (BOE)** BASED ON KNOWN METRICS. ONCE PROGRAM MANAGEMENT AGREES TO THE BUDGET YOUR TEAM WILL BE ALLOCATED, HOURS TO COMPLETE EACH TASK ALONG WITH THE CORRESPONDING BUDGET TO COMPLETE EACH TASK WILL BE ASSIGNED TO EACH TASK IN THE IMS.

2) REMEMBER FOR A PROGRAM TO BE SUCCESSFUL, YOU NEED TO **EFFECTIVELY MANAGE COST** BY USING A SIMILAR BUFFER STRATEGY AS WHEN YOU WERE DEALING WITH THE SCHEDULE.

- EFFECTIVE PROGRAMS WILL CREATE A BUDGET RESERVE BY ADDING PRODUCTIVITY CHALLENGES TO EACH ORGANIZATION.
- AS THE SPM, YOU WILL NEED TO DETERMINE IF YOU ARE GOING TO LEVEL THAT PRODUCTIVITY CHALLENGE EQUALLY ACROSS ALL DEVELOPMENT EFFORTS, SUPPORT, AND MANAGEMENT ACTIVITIES OR ONLY ON A SUBSET OF THE DEVELOPMENT EFFORT.

EVEN IF YOU ARE NOT FORCED TO TAKE A PRODUCTIVITY CHALLENGE, IT IS ALWAYS A GOOD IDEA TO CREATE AN INTERNAL BUDGET RESERVE THAT YOU CAN MANAGE SINCE THE **TEAM THAT IMPLEMENTS A CAPABILITY/FEATURE ULTIMATELY NEEDS TO BUY INTO THE BUDGET THEY HAVE BEEN ALLOCATED.**

SW PROJECT PLANNING – ACCOUNTING FOR RISKS

3) ACCOUNT FOR RISK IN YOUR COST BASELINE

IN YOU ARE FORCED TO TAKE A PROGRAM PRODUCTIVITY CHALLENGE ALWAYS ADD A PROGRAM RISK ASSOCIATED WITH THE PRODUCTIVITY CHALLENGE.

ANOTHER TYPICAL SOFTWARE RISK THAT SHOULD ALWAYS BE INCLUDED IS A SW SIZE RISK IF NEW TECHNOLOGY WILL BE USED AND YOUR METRICS MANY NOT ACCURATELY REFLECT THE EFFORT TO DEVELOP THE PRODUCT.

ANOTHER TYPE OF SOFTWARE RISK THAT SHOULD ALWAYS BE INCLUDED IS A RISK DEALING WITH CAPABILITIES/FEATURES THAT HAVE A PERFORMANCE REQUIREMENT ASSOCIATED WITH THEM THAT YOUR SYSTEM CURRENTLY DOES NOT MEET.

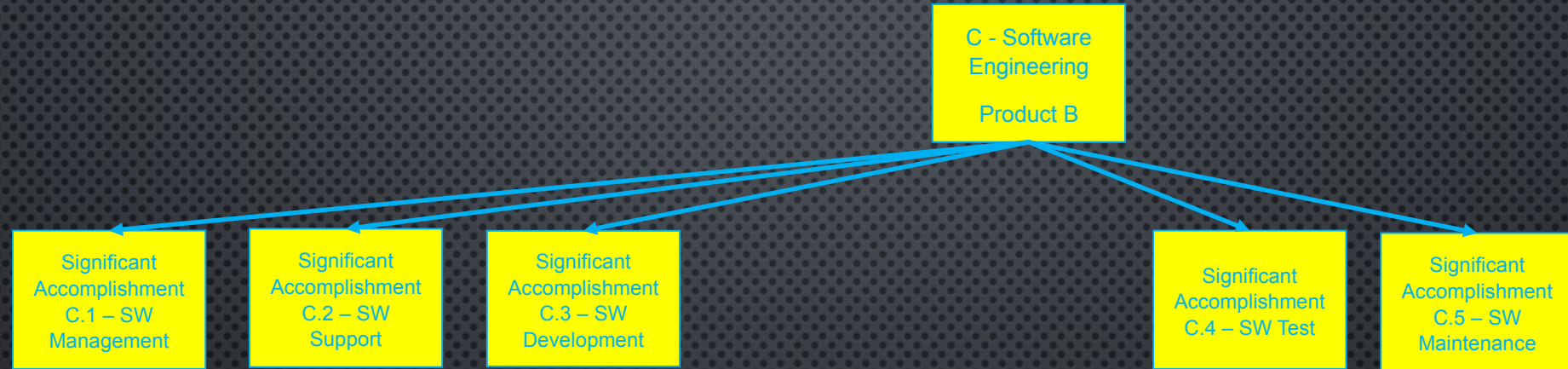
FINALLY, YOU SHOULD INCLUDE ANY RISKS DUE TO OBSOLESCENCE, HARDWARE AVAILABILITY, AND SUB-CONTRACTOR ISSUES.

IF THERE ARE HIGH RISK ITEMS THAT ARE A PART OF YOUR BID, YOU MAY WANT TO CONSIDER ASKING THE PROGRAM TO FUND A RISK MITIGATION ACTIVITY AS PART OF YOUR INITIAL SOFTWARE BASELINE.

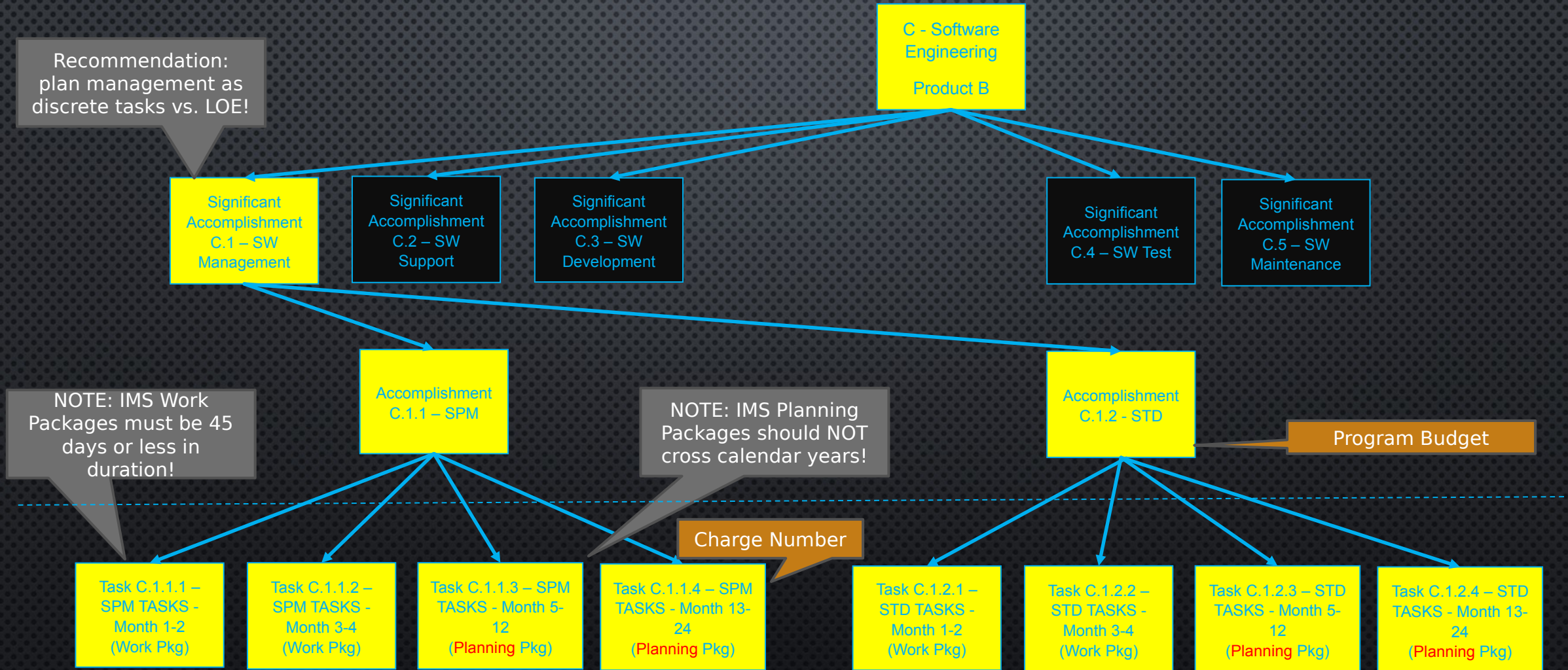
SW PROJECT PLANNING – DECOMPOSING THE WBS

4. AS PREVIOUSLY STATED, THE HIGH-LEVEL WBS IS TYPICALLY DEFINED AS PART OF THE PROPOSAL EFFORT IN RESPONSE TO A CUSTOMER RFP.
 - YOUR GOAL IS TO **CREATE** THE LOWER LEVEL **WBS STRUCTURE** SO THAT YOU CAN ACCURATELY **MANAGE THE HOURS AND ASSOCIATED COST** TO COMPLETE **EACH TASK** YOU ARE MANAGING.
 - THIS DATA WILL NOT ONLY HELP YOU MANAGE THIS PROGRAM BUT WILL PROVIDE YOUR ORGANIZATION WITH DATA THEY CAN USE WHEN BIDDING SW PRODUCTIVITY ON FUTURE PROGRAMS.
 - ALTHOUGH YOU ARE ULTIMATELY RESPONSIBLE FOR THE COST TO COMPLETE THE WORK YOU HAVE SIGNED UP FOR MY RECOMMENDATION IS TO ALWAYS MANAGE YOUR PROGRESS IN HOURS VS. DOLLARS. COMPLETING A TASK ON OR IN LESS HOURS THAN YOUR BASELINE HOURS WILL TELL YOU HOW WELL YOUR TEAM IS PERFORMING.
 - THE COST OF PERFORMING THE WORK IS DEPENDENT ON OTHER FINANCIAL FACTORS THAT MAY BE OUTSIDE OF YOUR CONTROL. BY UNDERSTANDING YOUR TEAM'S PERFORMANCE BASED ON HOURS, YOU WILL BE ABLE TO EXPLAIN YOUR TEAM'S PERFORMANCE BASED ON DOLLARS ONCE YOU GAIN AN UNDERSTANDING OF THOSE OUTSIDE INFLUENCES.

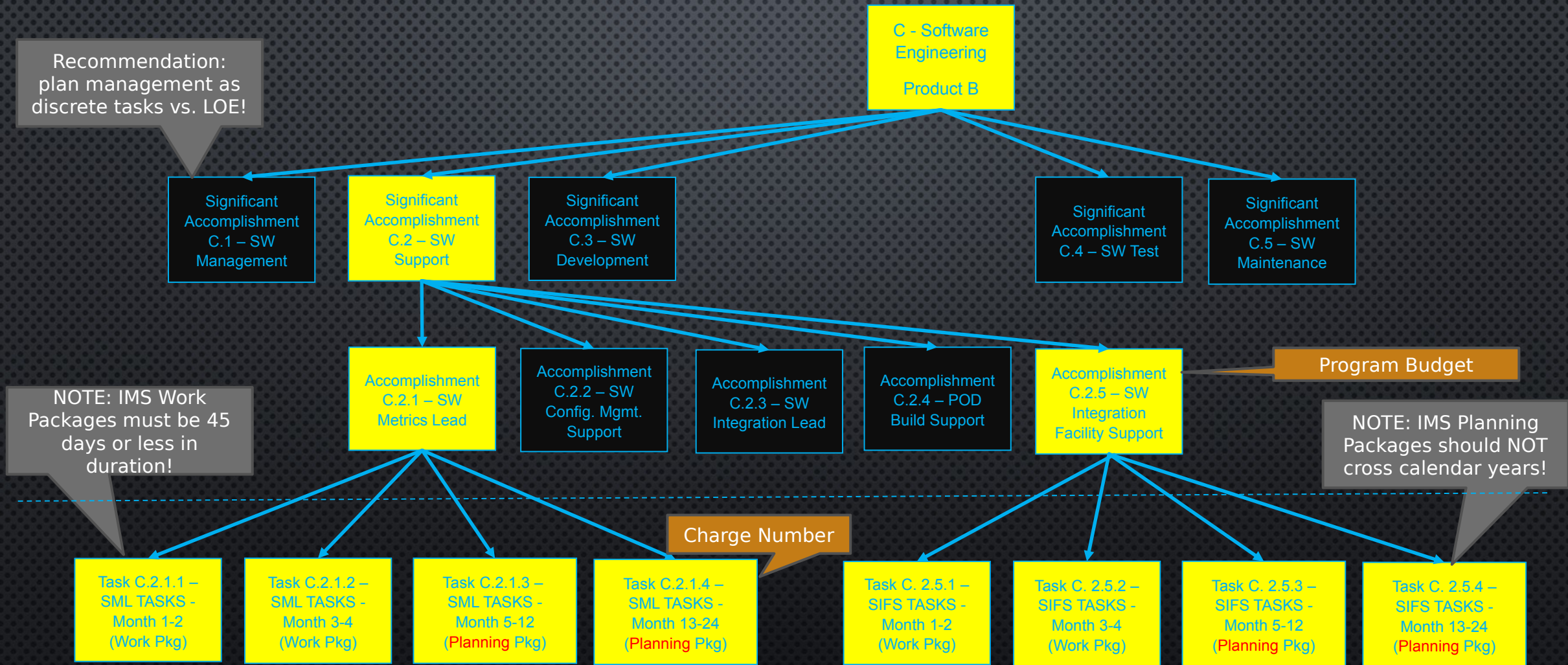
SW PROJECT PLANNING – DECOMPOSING THE WBS



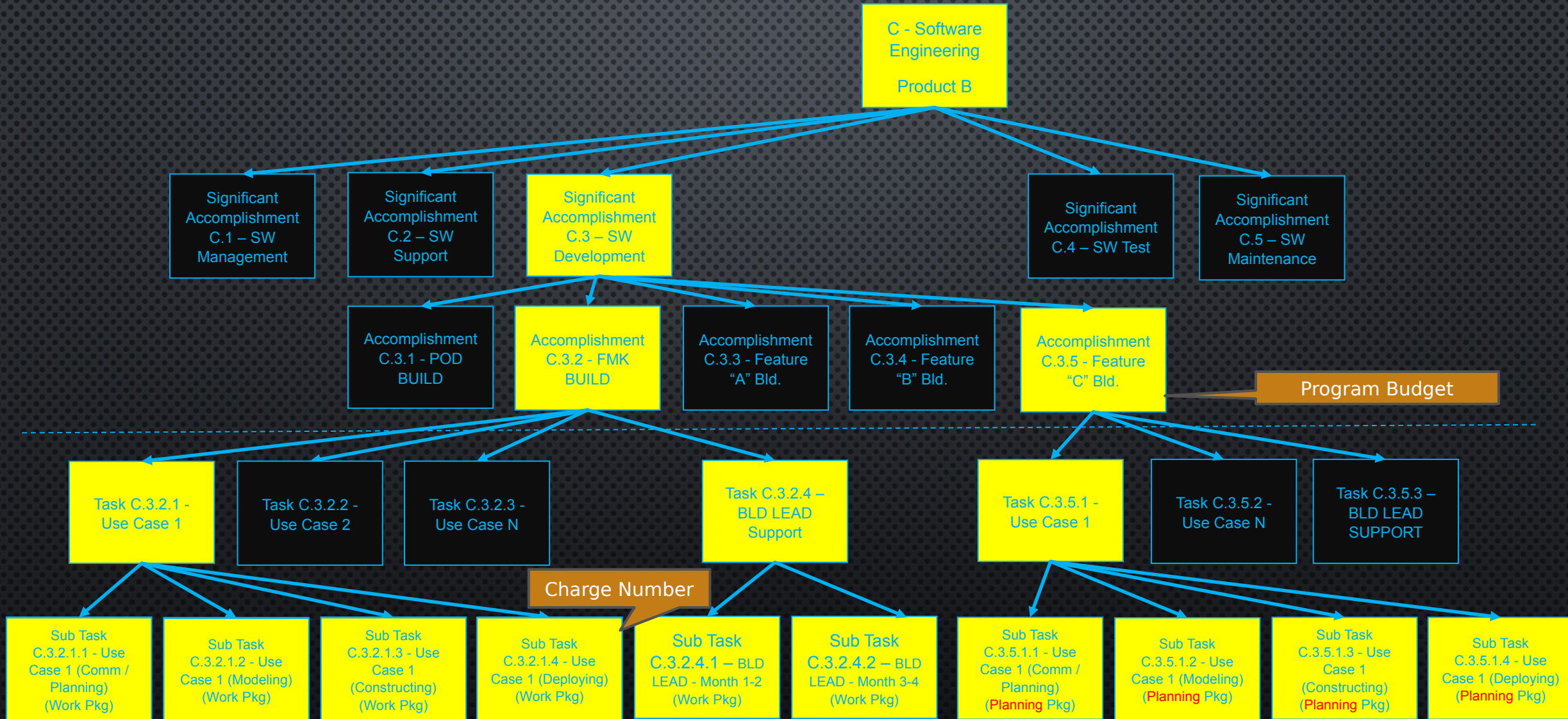
SW PROJECT PLANNING – DECOMPOSING THE WBS



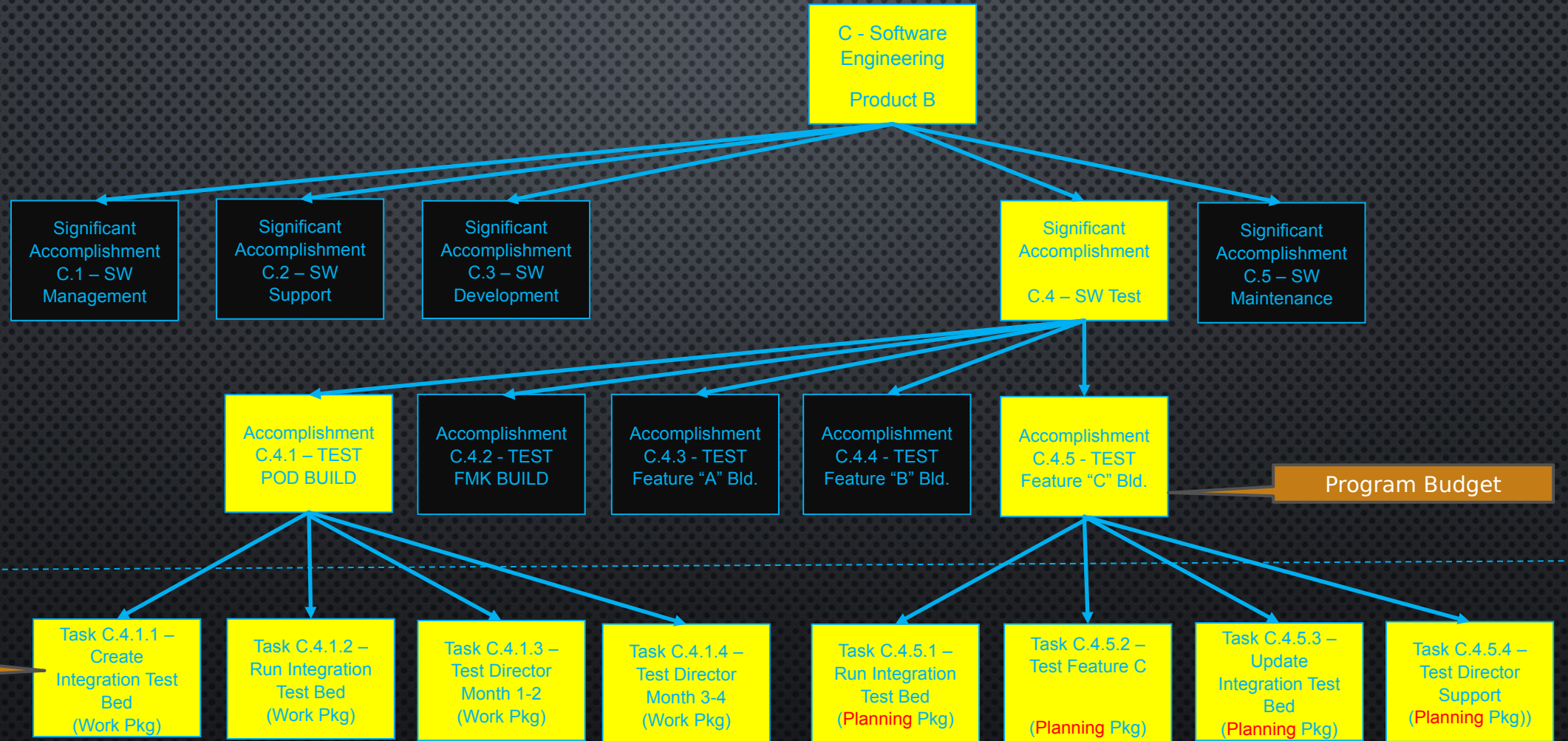
SW PROJECT PLANNING – DECOMPOSING THE WBS



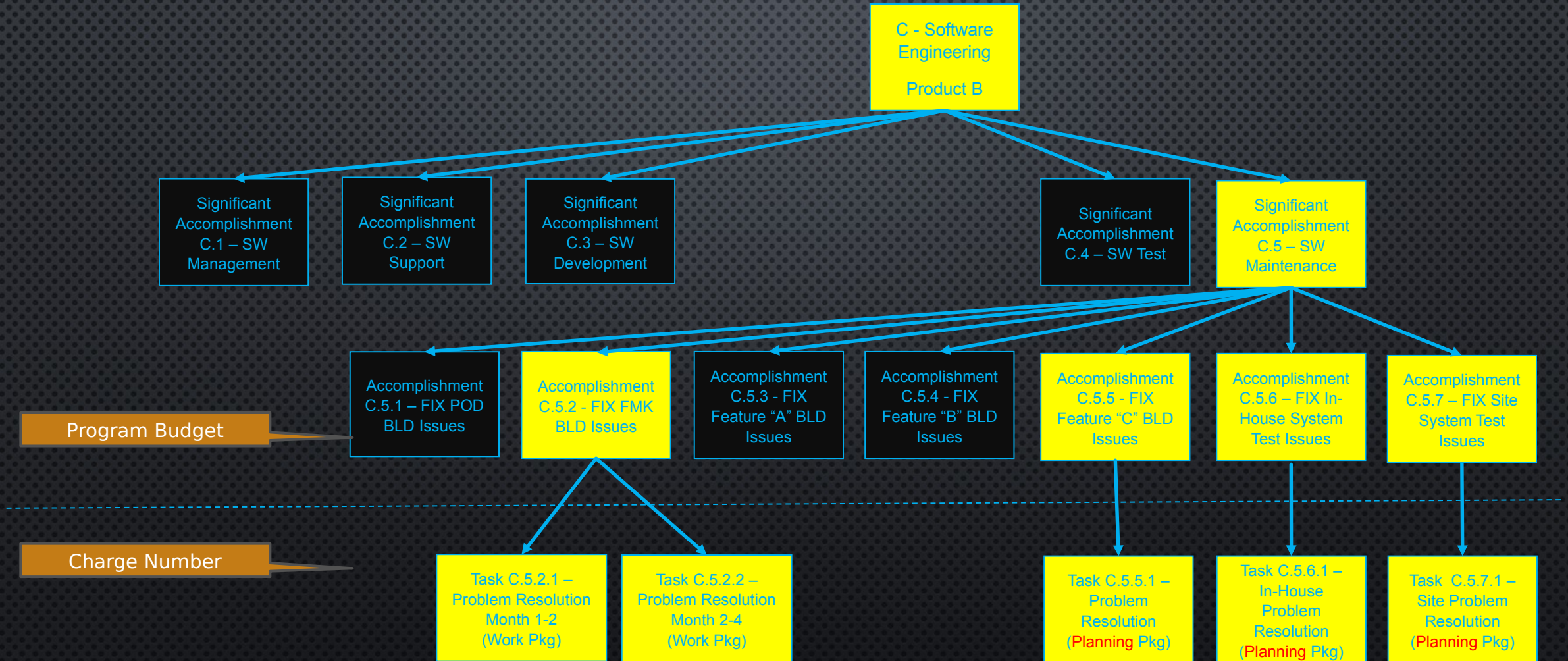
SW PROJECT PLANNING – DECOMPOSING THE WBS



SW PROJECT PLANNING – DECOMPOSING THE WBS



SW PROJECT PLANNING – DECOMPOSING THE WBS



SW PROJECT PLANNING – SDP

5. CREATING A SOFTWARE DEVELOPMENT PLAN (SDP)

ASIDE FROM DEVELOPING THE SBP, THE SDP IS THE REMAINING DOCUMENT THAT IS REQUIRED AS PART OF THE INITIAL SOFTWARE DEVELOPMENT PHASE.

THE SDP CAPTURES THE MANAGEMENT APPROACH AND ENGINEERING ENVIRONMENT EFFORT ASSOCIATED WITH THE PROGRAM. IT IDENTIFIES:

- THE SOFTWARE ORGANIZATIONAL STRUCTURE WITH RESPECT TO THE PROGRAM,
- THE HIGH-LEVEL SOFTWARE SCHEDULE WITH KEY MILESTONES,
- THE SOFTWARE RISKS ASSOCIATED WITH THE PROPOSED DEVELOPMENT,
- THE SOFTWARE MEASUREMENT AND ANALYSIS PLAN (TAKING NEW, MODIFIED, AND REUSE DEVELOPMENT INTO ACCOUNT),
- THE SOFTWARE CONFIGURATION MANAGEMENT PLAN (ENSURES ONLY TESTED SOFTWARE GETS CHECKED ONTO A BUILD),
- THE SOFTWARE ENGINEERING ENVIRONMENT (IN-HOUSE LAB FACILITIES),

THE SDP ALSO CAPTURES THE TECHNICAL APPROACH TO DEVELOPING THE SOFTWARE ASSOCIATED WITH A PROGRAM.

- THE SOFTWARE DEVELOPMENT PARADIGM THAT WILL BE FOLLOWED INCLUDING ANY STANDARDS,
- THE SOFTWARE PROCESSES THAT WILL BE FOLLOWED (THESE CAN BE SEPARATE DOCUMENTS THAT ARE REFERENCED) AND ARTIFACTS THAT WILL BE PRODUCED,
- THE USE AND INTEGRATION OF FREE AND OPEN-SOURCE SOFTWARE (FOSS) AND COMMERCIAL OFF-THE-SHELF (COTS) SOFTWARE,
- THE SOFTWARE BUILD PLAN (TYPICALLY SPECIFIED IN A SEPARATE DOCUMENT).

SW PROJECT PLANNING – ORGANIZATION

PROGRAM ORGANIZATIONAL STRUCTURED

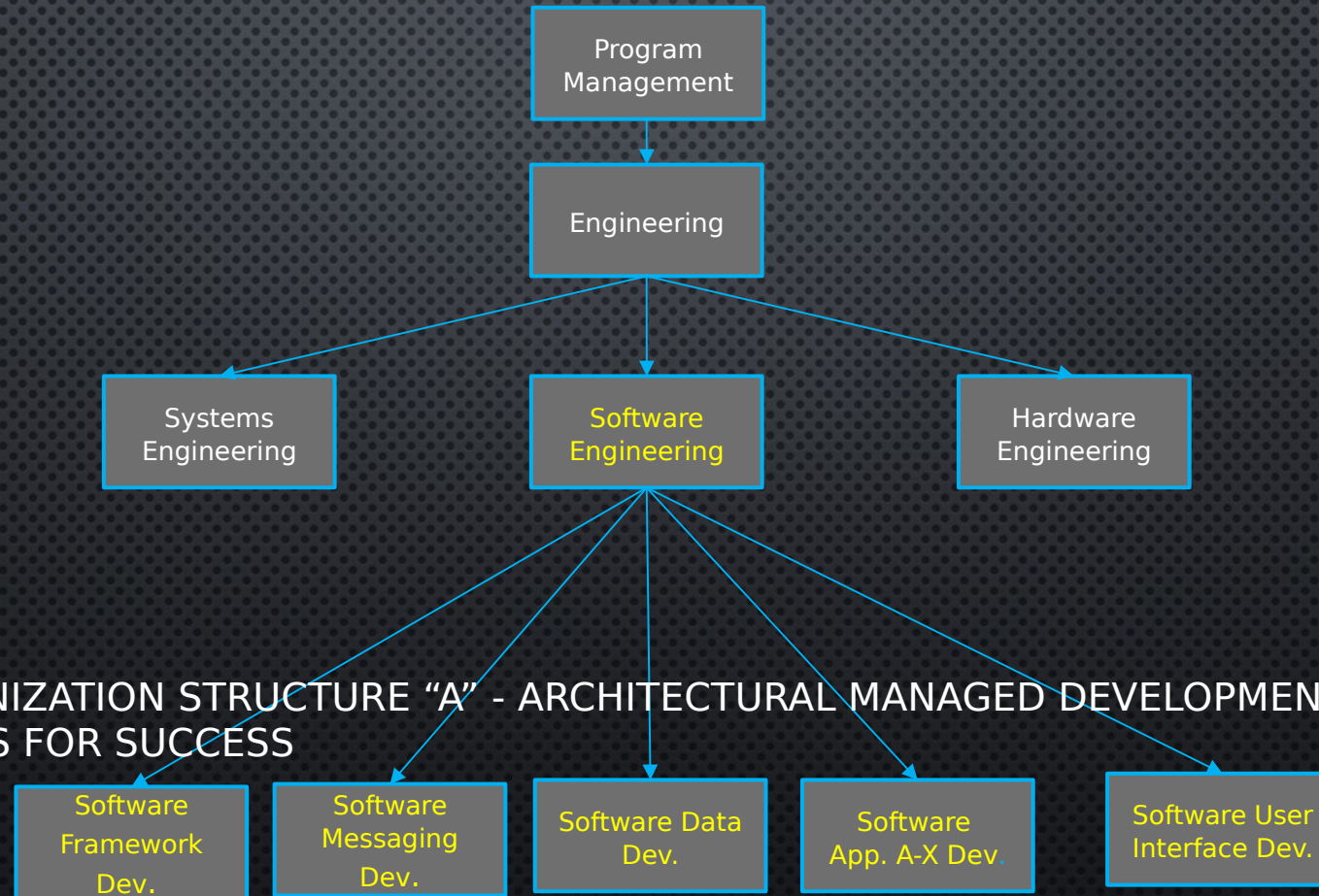
THE ORGANIZATIONAL STRUCTURE OF THE PROGRAM IS TYPICALLY DETERMINED AT THE TIME THE PROGRAM IS BID SINCE IT NEEDS TO ACCOUNT FOR THE USE OF AN INTEGRATED PRODUCT TEAM (IPT) DEVELOPMENT STRUCTURE VS. A NON-IPT STRUCTURE.

PROGRAM STRUCTURE HAS A DIRECT IMPACT ON YOUR ORGANIZATION'S ABILITY TO IMPLEMENT THE ITERATIVE OR AGILE APPROACH SPECIFIED IN THE SDP DOCUMENT.

FOR EXAMPLE, A PROGRAM STRUCTURE THAT IS ORGANIZATIONAL OR ARCHITECTURAL BASED VS. USING AN IPT STRUCTURE IS MORE LIKELY TO HAVE LIMITATIONS IN RECEIVING ALL OF THE BENEFITS OF THE ITERATIVE OR AGILE APPROACH THE SW TEAM IS IMPLEMENTING AS SHOWN IN THE DIAGRAMS BELOW.

PROGRAM ORGANIZATION STRUCTURE "A"

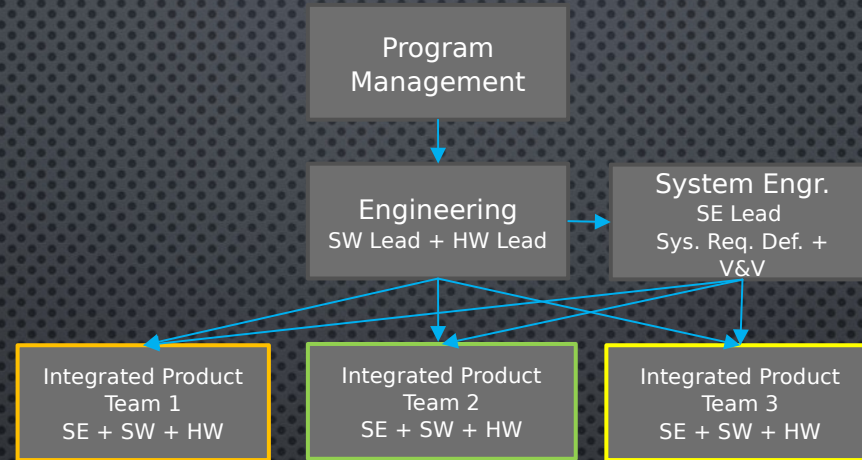
DISCIPLINE BASED ORGANIZATIONAL STRUCTURE - PROGRAMS AND DISCIPLINES HAVE THEIR OWN GOALS FOR SUCCESS



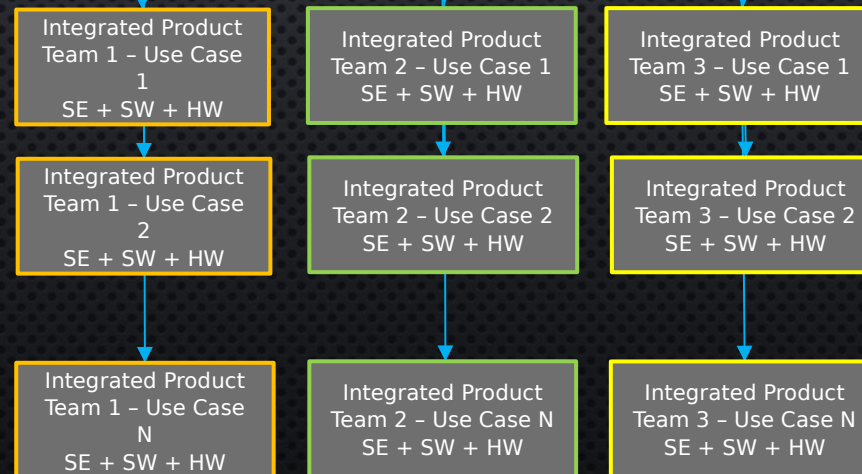
SOFTWARE ORGANIZATION STRUCTURE "A" - ARCHITECTURAL MANAGED DEVELOPMENT TEAMS HAVE THEIR OWN GOALS FOR SUCCESS

PROGRAM ORGANIZATION STRUCTURE "B"

INTEGRATED PRODUCT TEAM ORGANIZATIONAL STRUCTURE - BALANCES PROGRAMS AND GOALS FOR SUCCESS



SOFTWARE ORGANIZATION STRUCTURE "B" - USE CASE DRIVEN DEVELOPMENT TEAMS CAN MORE EASILY FOLLOW AN ITERATIVE AND EVOLUTIONARY PROCESS MODEL.



SW PROJECT PLANNING – CONFIGURATION MGMT.

PROGRAM SW CONFIGURATION MANAGEMENT PLAN

THE SW CONFIGURATION MANAGEMENT PLAN DOCUMENTS HOW SOFTWARE ARTIFACTS ARE CONTROLLED AND HOW CHANGE TO THESE ARTIFACTS IS MANAGED THROUGHOUT THE EXECUTION OF THE PROGRAM. THIS INCLUDES REQUIREMENTS TRACEABILITY, DESIGN DOCUMENTATION, CODE, TEST DOCUMENTATION, SOFTWARE PROCESS DOCUMENTS, ETC.

THE SW CONFIGURATION MANAGEMENT PLAN MAY ALSO SPECIFY SPECIFIC TOOLS USED TO CONTROL THE CONFIGURATION OF THE REQUIREMENTS AND CODE IN PLANT AS WELL AS AT THE CUSTOMER FACILITY.

SW PROJECT PLANNING – INTEGRATION FACILITY

PROGRAM SW INTEGRATION AND TEST ENVIRONMENT

THE ONLY WAY TO ENSURE A PRODUCT WORKS AS SPECIFIED IS TO TEST IT IN ACTUAL ENVIRONMENT IN WHICH IT WILL BE USED. UNFORTUNATE SOFTWARE ERRORS FOUND IN THE ACTUAL DELIVERABLE ENVIRONMENT ARE THE MOST EXPENSIVE TYPES OF ERRORS TO FIX.

THE SOONER A SW ERROR CAN BE DETECTED AND FIX DIRECTLY CORRELATES TO COST ASSOCIATED WITH FIXING THE ERROR.

- ERRORS DISCOVERED AND FIXED DURING UNIT TEST ARE THE CHEAPEST ERROR TO FIX.
- ERRORS DISCOVERED DURING INTEGRATION/SYSTEM TESTING THAT ARE FOUND/FIXED IN THE INTEGRATION AND TEST ENVIRONMENT END UP BEING MORE EXPENSIVE TO FIX SINCE THEY IMPACTS A LARGER NUMBER OF INDIVIDUALS AND REQUIRES A NUMBER OF TEST ASSETS IN THE INTEGRATION FACILITY TO VERIFY THE ISSUE WAS FIXED. HOWEVER, THE COST OF FIXING A SOFTWARE ERROR PRIOR TO THE PRODUCT LEAVING THE SITE IS KEY TO MINIMIZING PROGRAM COSTS FOR PROBLEM RESOLUTION.

SW PROJECT PLANNING – INTEGRATION FACILITY

HENCE IT IS CRITICAL THAT A PLAN IS PUT IN PLACE DURING PROGRAM STARTUP THAT CREATES AN INTEGRATION TEST ENVIRONMENT THAT ACCURATELY REFLECTS THE TARGET ENVIRONMENT AND HAS ENOUGH EQUIPMENT TO SUPPORT THE DEVELOPMENT SCHEDULE DEFINED BY THE PROGRAM.

- HARDWARE COMPONENTS THAT NEED TO BE CONSIDERED INCLUDED: PHYSICAL SPACE, NUMBER OF WORKSTATIONS AND SERVERS THAT COMPRISE A STRING, NUMBER OF STRINGS THAT COMPRISE A SITE, NUMBER OF SITES AT THE TARGET DOMAIN.
- SCHEDULING COMPONENTS THAT NEED TO BE CONSIDERED INCLUDES AVERAGE AND PEEK UTILIZATION OF THE EQUIPMENT.

SW PROJECT PLANNING – SW PROCESSES

SOFTWARE PROCESS THAT WILL BE FOLLOWED DURING THE EXECUTION OF THE PROGRAM CAN EITHER BE DIRECTLY CONTAINED IN THE SDP OR DEFINED IN SEPARATE PROGRAM DOCUMENTS.

EXAMPLES OF SOFTWARE PROCESS THAT WOULD BE REFERENCED WITHIN THE SDP ARE HIGHLIGHTED IN THE DIAGRAM TO THE RIGHT.

NOTE: SOMETIMES IT IS CHEAPER TO INCLUDE A COTS PRODUCT AS PART OF THE SOLUTION TO THE SYSTEM BEING DESIGN OR TO USE A FOSS PRODUCT. IF THAT SITUATION, IT IS IMPORTANT TO ADDRESS HOW THE COTS/FOSS PRODUCT WILL BE INTEGRATED, TESTED, AND CONTROLLED AS PART OF THE SOFTWARE DEVELOPMENT PLAN.

SOFTWARE ENGINEERING DISCIPLINE

SOFTWARE ORGANIZATION

KEY PROGRAM 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

REFERENCES

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

MITRE SYSTEMS ENGINEERING GUIDE

[HTTPS://WWW.MITRE.ORG/PUBLICATIONS/SYSTEMS-ENGINEERING-GUIDE/ACQUISITION-SYSTEM-S-ENGINEERING/ACQUISITION-PROGRAM-PLANNING/INTEGRATED-MASTER-SCHEDULE-IMSINTEGRATED-MASTER-PLAN-IMP-APPLICATION](https://www.mitre.org/publications/systems-engineering-guide/acquisition-system-s-engineering/acquisition-program-planning/integrated-master-schedule-imsintegrated-master-plan-imp-application)

PRINCIPLES OF SOFTWARE ENGINEERING MANAGEMENT, BY TOM GILB, COPYRIGHT 1988 BY ADDISON-WESLEY PUBLISHING COMPANY, NEW YORK, NY

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