Table of Contents

1: Demystifying Process Tailoring: Creation Unified Process (CUP)

Create, use and distribute your own professional software delivery process variant. Start by quickly getting acquainted with the following two sources of information on the web:

- OpenUP ➔ http://epf.eclipse.org/wikis/openup/ [1]
- Creation Unified Process ➔ https://ateliermdev.azurewebsites.net/ [2]

Expand **Delivery Processes** in the left-hand menu to access work breakdown structures. Don't get tangled up in the details. Focus on the nature, structure, composition and synthesis of the objects (quickly) before proceeding. You'll have a chance to roll up your sleeves soon.

2: Overview

Software delivery projects span a wide variety of technology domains, customers, application types, size and complexity. No single standard is able to cater to a wide variety of project types. This learning object is a process tailoring accelerator and provides guidance to produce a fit for purpose delivery process variant for any software delivery initiative.

Quite simply, 'Process Tailoring' is to alter or adapt a process to meet an initiative's objectives and environment. [3][Ref CMMI – Section 4.2: Process Tailoring]

Purpose of process tailoring

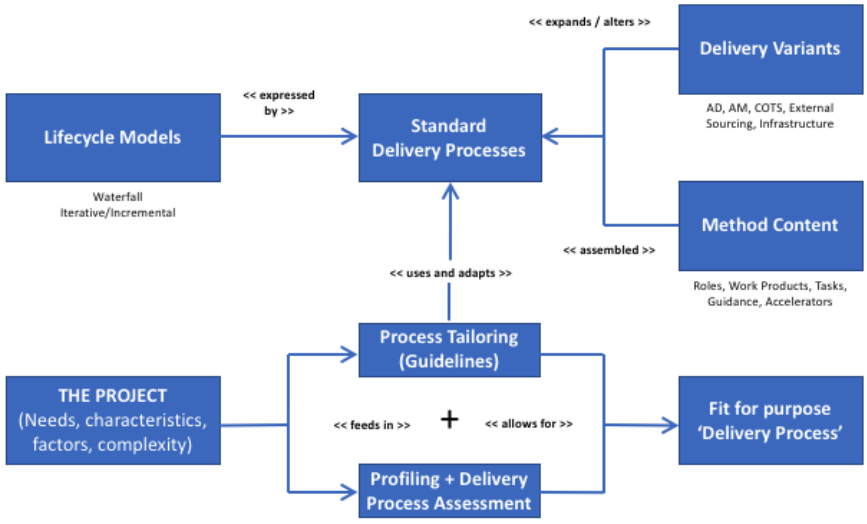
- To right-size the delivery process according to the specific needs of a project
- To provide enough relevant guidance for the project members to do their jobs effectively
- To provide relevant and accessible process descriptions for the members of the project

How?

- A delivery process is customized to meet those needs – through adaptation or alteration
- Tailoring guidelines describe what can and cannot be modified against a set of standard delivery variants

What do you need? (Inputs)

- A standard set of dynamic and tailorable process assets as a baseline. Derived from the baseline, you'll need (and be able to create)
    - Defined Lifecycle models
    - Defined Delivery process variants



Note: This figure is derived from a picture used to promote process tailoring in an organizational setting [4]

3: Revelations - The maturity and make work lies

Agree that there is an abundance of processes and project management tools and techniques, they're just that tools, nothing more; each a revenue generating machine hidden behind the lie of certification and maturity. The truth is, it's all much simpler than it seems.

- These baselines, such as PMBOK, DAD, Prince2, RUP, OpenUP, are tools, or more accurately communication tools that facilitate interoperability between practitioners and or organizations; we point to these baselines as a basic form of communicating the process of delivery; more importantly, we actually tailor them every day
- When push comes to shove, do you start from scratch? No. We rely on what we've learned, whether through past experiences, education, training and or a composite of all to make things happen
- Process tailoring happens every day, in fact, probably every moment. Small to medium sized organizations, such as start-ups, rely on process tailoring to deliver, we just don't realize that is exactly what we're doing, tailoring our inherent process experience to deliver
- Simply, we don't document the process and subject it to continuous improvement

In essence, Process Tailoring, is the ultimate form of **allusion** and improvisation in the context of project delivery. Process assets, are in fact, the props that practitioners use while improvising delivery. By formally documenting these assets, one can quickly scale the software engineering maturity mountain quite quickly. The tools and information to do so are out there and readily available. The following modules provide an example of leveraging accessible delivery management process assets and tailoring them to suit your specific needs.

4: Allusions

What is an allusion?

- Oxford says: "An expression designed to call something to mind without mentioning it explicitly; an indirect or passing reference" [5]
- Merriam-Webster says:
    - "an implied or indirect reference especially in literature a poem that makes allusions to classical literature; also : the use of such references" [6]
    - "the act of making an indirect reference to something : the act of alluding to something"

In thinking about allusions in the context of theology and religion, specifically the context of the prophecy of 'Revelations', famed theologian "G.K. Beale sought to develop a system that distinguished 'clear', 'probable' and 'possible' allusions:

- A **clear** allusion is one with almost the same wording as its source, the same general meaning, and which could not reasonably have been drawn from elsewhere.
- A **probable** allusion contains an idea which is uniquely traceable to its source.
- **Possible** allusions are described as mere echoes of their putative sources" [7]

5: Measuring Allusions: One metric to rule them all

Metrics today are in abundance, there's a metric for anything and everything. What's lacking is a polymorphic, abstract, multi-dimensional, multi-disciplinary and cross functional metric that rules them all. Building on the previous definition and described system or pattern of allusion (possible ➔ probable ➔ clear), the following posits the creation of a metric to rule them all.

The proposed metric is based on our fundamental desire to evolve. It is an open-ended measure of progression expressed as a simple numeric pattern that we are all familiar with and can relate to. It notionally associates itself to the aforementioned system for distinguishing allusions:

-1	**0**	**1**
Possible	**Probable**	**Clear**
The unrealized idea or concept, state of ideation	*The idea as a work in progress (a project)*	*The realized idea, product or deliverable*

This is an extremely high-level standard that can be modified concretely or abstractly to align with any application or level of business practice or even use in every-day life. Its fundamental values are black and white:

- Its ability to be highly polymorphic while traceable; easy for rapid roll up reporting of various classes of data or information
- Flexible so that the most basic forms of measurement, analysis and or statistical tools can be applied against it given that it implicitly defines an upper and lower specification limit; great for control
- Its ability to organically classify the dimension of time at its highest possible level of standard, basically ➔ Past, Present and Future and for that to be aligned to any form or system of classification

6: One metric to rule them all: Sample measures

The table below provides examples of this metric system's polymorphic traits:

Measure	**(-1) Past**	**(0) Present**	**(+1) Future**
Possible classification(s)			
Ideas	Possible	Probable	Clear
Time	Yesterday	Today	Tomorrow | Day 1
Finance	Loss	Break even	Profit
Delivery / Project Management	Closure/Lessons Learned	Planning & Initiation | Active Sprint	Execution, Build & Deployment
Assets	Legacy	Current	Business & Technology Innovation

7: Process tailoring is an exercise in creativity, over maturity

No one is arguing that process tailoring is a continuous improvement technique that comes with a high level of maturity in the software development domain. The argument, in fact, is that standardizing the continuous improvement and creative aspect of process tailoring as soon as possible drives rapid excelling in maturity. It is the ultimate form of self-discipline and provides the tool kit to imminently capture the maturity process as it unfolds. The following content is an attempt to demonstrate how simple and fun process tailoring can actually be. One can argue for fun that process tailoring is actually a form of magic realism. [8]

The product is an improvised delivery framework, inspired by the OpenUP process and it is called Creation Unified Process (CUP). Its core objectives and themes are the integration of storytelling and project delivery. CUP begins by adapting improvisation principles, guidelines and rules as its most basic form of inspiration, the means of fusing the practical with the fantastical.

8: Improvisation and adaptation as inspiration for Process Tailoring

Trying to describe how relevant improvisation and adaption is to process tailoring is a relatively subjective task. The only way to demonstrate this is by immersed example. The balance of this module is **adapted** from several sources as a means of demonstrating how powerful these techniques (improvisation and adaption) are when applied. [9] [10] [11] [12]

What is Creation Unified Process?

In CUP, **storytellers** create products in which iterations (short or long) are interrelated by stories/ideas/needs, requirements, or themes, all integrated into a project lifecycle. The delivery lifecycle may take the form of an existing type delivery SDLC lifecycle such as Waterfall or Agile and or a logical combination of different lifecycle methods and variants.

Creation Unified Process (CUP) is a baseline of delivery process assets for delivery of products or services based on direction from customers (or end users), delivery practitioners and or stakeholders. CUP is a form of delivery where most or all of what is created is created at the moment it is delivered, a.k.a. improvisation. In its purest form, the story, the actors and the product are created collaboratively by delivery practitioners functioning as **storytellers.** As the improvised delivery unfolds in present time, mostly, without the use of formalized requirements or design. Design never stops.

Developing the Storyboard (and or profiling the project)

The Guiding Principles of Creation Unified Process

Listen: easier said than done, and that's exactly the point
Agreement: say yes and add something, don't reject ideas
Team Work: have a group mind, think of others
Don't Block: stealing time / not listening / changing topic
Relationship: focus on ALL available connections and traceability, not just the subject of delivery
Initiation: who, what, when to set up for delivery
Point of View, Opinion and Intention: have them, these helps express and build your character
Be in Role and Function: maintain your role and function throughout the delivery lifecycle
Don't Ask Questions: too many questions can make your partners do all the work
Make Active Choices: Do something. Don't be talking heads

Applying CUP principles in the workplace and life

Guiding principles of CUP are useful, not just in the workplace, but in everyday life. For example, Stephen Colbert in a commencement address said: Well, you are about to start the greatest improvisation [**story**] of all. With no script. No idea what's going to happen, often with people and places you have never seen before. And you are not in control. So say "yes." And if you're lucky, you'll find people who will say "yes" back.

There has been much interest in bringing lessons from improve [and storytelling] into the corporate world. In a New York Times article titled "Can Executives Learn to Ignore the Script?", Stanford professor and author, Patricia Ryan Madson notes, "executives and engineers and people in transition are looking for support in saying yes to their own voice. Often, the systems we put in place to keep us secure are keeping us from our more creative selves." ***Tina Fey in her book Bossypants lists several rules of improv that apply in the workplace.***

4 Key Rules Explained

Rule #1: Agree

The first rule of CUP is AGREE. Always agree and SAY YES. When you're improvising [and storytelling], this means you are required to agree with whatever your partner has created.

Lesson: *Respect What Your Partner Has Created*

Rule #2: Not Only Say Yes…Say Yes And

The second rule of CUP is not only to say yes, but YES, AND. You are supposed to agree and then add something of your own. Don't be afraid to contribute. It's your responsibility to contribute. Always make sure you're adding something to the discussion. Your initiations are worthwhile.

Lesson: *Contribute Something*

Rule #3: Make Statements

The next rule is MAKE STATEMENTS. This is a positive way of saying "Don't ask questions all the time." If we're in a meeting and I say, "Who are you? Where are we? What are we doing here? What's in that box?" I'm putting pressure on you to come up with all the answers. In other words: Whatever the problem, be part of the solution. Don't just sit around raising questions and pointing out obstacles. Make statements, with your actions and your voice. Instead of saying "Where are we?" make a statement like "Here we are in Spain, Dracula."

Lesson: *Don't Ask Questions All The Time*

Rule #4: There are No Mistakes…Only Opportunities

Okay, "Here we are in Spain, Dracula" may seem like a terrible start to a delivery process, but this leads us to the best rule: THERE ARE NO MISTAKES, only opportunities. In CUP there are no mistakes, only beautiful happy accidents. And many of the world's greatest discoveries have been by accident.

Lesson: *Stay Positive, Learn to Adapt*

Final word on principles and rules

In delivery (and business), it pays to have the qualities of an improviser [**and storyteller**]

- We all have behavior that comes naturally to us
- We might like to take things slow, mingle with friends, or have alone time
    - And it's not always advantageous to behave this way at work.
    - So, we adapt and improvise.
- We accept things that come our way…even though we don't like it
- We add our personal touch as projects come our way…to make work more enjoyable
- We make mistakes…and learn to roll with it

Structure and Process

CUP often allows an interactive relationship with the customer or end user. Delivery practitioners frequently solicit suggestions from the customer or end user as a source of inspiration, a way of getting the customer involved, and as a means of proving that the product is not subjective, but truly represents the needs and ideas of the customer or end user.

In order for CUP to be successful, the delivery practitioners involved must work together responsively to define the parameters and action of the delivery process and product, in a process of co-creation. With each spoken word or action in the delivery process, a delivery practitioner makes an offer, meaning that he or she defines some element of the reality of the product (or service) being delivered. This might include:

- Create requirements or design
- Profile a project
- Create a delivery environment or test data emulating the real environment

These activities are also known as endowment. It is the responsibility of the other delivery practitioners to accept the offers that their fellow practitioners make; to not do so is known as blocking, negation, or denial, which usually prevents the product from developing. Some practitioners may deliberately block (or otherwise break out of character) for a grandstanding effect—this generally prevents delivery from advancing and is frowned upon by many practitioners. Accepting an offer is usually accompanied by adding a new offer, often building on the earlier one; this is a process improviser refer to as "Yes, And..." and is considered the cornerstone of improvisational delivery technique **[or process tailoring].** Every new piece of information added helps the practitioners to refine their role and function to progress the action of delivering the product. The "Yes, And..." rule, however, applies to the early stages of delivery since it is in this stage that a "base (or shared) reality" is established in order to be later redefined by applying the "if (this is true), then (what else can also be true)" practice progressing delivery into the realization of the product.

The unscripted nature of CUP and process tailoring also implies no predetermined knowledge about the process assets that might be useful in delivery. Organizations may have at their disposal some number of readily accessible process assets that can be called upon at a moment's notice, but many delivery practitioners eschew process assets in favor of the infinite possibilities available through CUP and or process tailoring. That being said, having a diverse understanding of basic delivery lifecycle models and variants such as Waterfall, Agile, PMBOK, Prince, DAD, RUP, OpenUP, as well as a well-documented and optimized organizational baseline of process assets to spring from facilitates the communication process between practitioners. This is absolutely necessary because delivery practitioners may be required to play a variety of roles without preparation, they need to be able to construct

solutions quickly with varying techniques as demanded by the situation. The delivery practitioner may be called upon to perform differing role functions and or tasks. Being capable of quickly adapting to new objectives and or expectations is an important part of CUP.

The CUP baseline of process assets, provide any delivery practitioner with a comprehensive understanding at all times of the most basic aspect of what is expected during delivery, inclusive of detailed relationships between process assets. As objectives or motivations may change during delivery, a practitioner can always quickly refer to the baseline for inspiration on how to act as to meet the highest level of delivery standards while simply documenting variances from the standard baseline, thus tailoring to adapt to the objectives that they believe their project seeks.

9: Implement Process Tailoring in 10 Steps

STEP 1: Identify your baseline

Start with a well-known and stable project baseline or work breakdown structure:

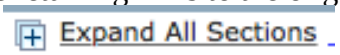
- Recommending use of OpenUP. Why?
    - It includes all of the process assets required for comprehensive delivery of a project, including:
        - Activities
        - Roles
        - Work Products / Deliverables
        - Artifacts
    - All relationships between process assets are clearly described
    - It is open and accessible to the public
    - As a component of the Eclipse Process Framework, it is tailorable using the Eclipse Process Composer
    - While it is an "iterative and incremental" delivery lifecycle, it can be tailored into a wide variety of delivery lifecycle models or variants (including Waterfall and or Iterative/Incremental/Agile)

STEP 2: Start developing an excel workbook to document your tailoring effort

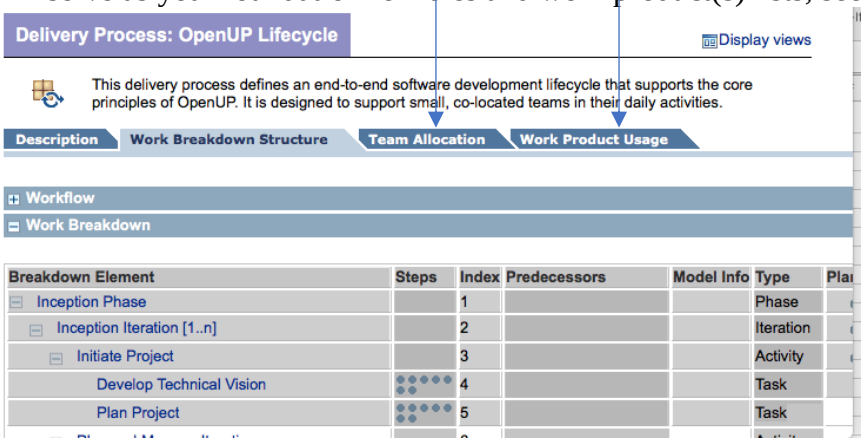
- Develop an excel workbook which documents the entire effort from start to finish, this will ultimately be your ➔ **Delivery Workbook (or Playbook)**
- Use Excel, everybody has access to it one way or another.
    - **See CUP Workbook**

STEP 3: Document the baseline

- Document the base that is inspiring the development of your own delivery process assets baseline
- Note: The benefits of using OpenUP is that you can copy the WBS (and more) from the OpenUP Delivery Process lifecycle page ('Expand All Sections' first) directly into excel and get started instantly, while retaining links to the original context



- Repeat this step for the Team Allocation and Work Product Usage sections, which will serve as your foundation for roles and work product(s) lists; see STEP 6



  - **See CUP Workbook: Sheet REFA-OUP Base [Col A, B, D]**

	A	B	D
	OUP Index	Open Unified Process	Type
	1	Inception Phase	Phase
	2	Inception Iteration [1..n]	Iteration
	3	Initiate Project	Activity
	4	Develop Technical Vision	Task

STEP 4: Document a first cut of your adapted process

- This should be a 1:1 adaptation from your chosen base and does not include any additions or subtractions in process assets, only modify the description of the phase, activity or task name and
- Continuously maintain a 1:1 traceable relationship with the baseline
  - **See CUP Workbook: Sheet REFA-OUP Base [Col C, D]**

	A	B	C	D
1	OUP Index	Open Unified Process	Creation Unified Process Track Map	Type
2	1	Inception Phase	Inception Phase	Phase
3	2	Inception Iteration [1..n]	Inception Iteration [1..n]	Iteration
4	3	Initiate Project	Initiate CUP Track	Activity
5	4	Develop Technical Vision	Develop Storyboard	Task
6	5	Plan Project	Plan Track	Task

STEP 5: Document all modifications and additions to phases, activities or tasks

- Using the outcomes of STEP 3, create a separate sheet and document all modifications and additions to phases, activities or tasks
- Ensure to track major variances from the base with indicator(s) of your choice
- Make sure to not delete any rows and always maintain traceability to the base index; you can hide the column that contains the base info, but it is highly recommended that it is never deleted as a form of audit, proof of work and measurement of variance(s)
  - **See CUP Workbook: Sheet REFB-CUP Baseline**

	A	C	D	E	F	Q
1	OUP Index	CUP Index	Creation Unified Process Track Map	Type	Steps	
2	1	1	Inception Phase	Phase		
3	2	2	Inception Iteration [1..n]	Iteration		
4	3	3	Initiate CUP Track	Activity		
5	-	4	Define the Vision	Task	7	
6	4	5	Develop a Storyboard	Task	6	
7	-	6	Create Storyboard Profile	Task		

REFA-OUP Base | REFB-CUP Baseline | +

  - ***Note:*** *Indicators to track variances* **[Col A & Q]**

STEP 6: Create a list of roles and work products (deliverables)

- Do not start from scratch in developing these lists. Keep it simple. You can quickly start by using the OpenUP work products and roles and then augment, append and modify the list(s) based on unique operating practices and needs.
- Recommended to start with the Team Allocation breakdown offered by OpenUP, this can serve as a base for developing estimation(s).
    - See STEP 3
    - **See CUP Workbook: Sheet REFC-Allocation & Estimation [Col H]**

STEP 7: Author and publish delivery process wireframe(s) using Eclipse Process Composer

- Using the material developed via STEPS 1 through 6, document and publish delivery process wireframe(s) without detailed content, focus on developing method content / process asset(s) placeholders and creating wireframe work breakdown structure(s).
- Proceed to Modules **10** & **11** for how to author and process using Eclipse Process Composer; then return and proceed to STEP 8

STEP 8: Develop Allocation and Estimation Tool

Using the published wireframe baseline delivery process:

- Copy the Work Breakdown Structure to an excel spreadsheet and make any necessary changes to align the content as best as you can with the excel Workbook / Playbook you are continuously developing.
    - If you have the patience, maintain hyperlinks to web content if the wireframe has been published online
        - **See CUP Workbook: Sheet REFB-CUP Baseline**
    - Add all roles as columns twice.
        - The first set to display effort in days by role **[Col Q-AG]**
        - The second set to display effort cost by role. **[Col AI-AX]**
- Copy the Team Allocation breakdown to an excel spreadsheet and make any necessary changes to align the content as best as you can with the excel Workbook / Playbook you are continuously developing.
    - Using formulas and fixed cells as much as possible, apply estimates using your organization's unit(s) of measure.
        - Apply **day** effort estimates to each role by task
        - **See CUP Workbook: Sheet REFC-Allocation & Estimation**

Role	Details	Model Info: Functio	Effort Estimate (Days)	Type
Analyst	Define the Vision	Performs as Owner	3	Task
Analyst	Detail Use-Case Scenarios	Performs as Owner	5	Task
Analyst	Outline Solution Options	Performs as Owner	4	Task
Analyst	Assess Solutions and Estimation	Performs as Additional	2	Task
Analyst	Assess Storyboard Profile	Performs as Additional	0.25	Task
Analyst	Create Test Cases	Performs as Additional	1	Task

        - Copy each individual effort estimate per role by task to the Baseline delivery process, using fixed cell formulas
        - **See CUP Workbook: Sheet REFB-CUP Baseline [Col Q-AG]**

Creation Unified Process Track Map	Type	Steps		Analyst	Any Role SME	Architect	CM Spec	Depl Eng	Designer	Developer
Outline Solution Options	Task	7		4	1	2			1	0.5
Assess Solutions and Estimation	Task	4		2	1					
Detail Use-Case Scenarios	Task	4		5		0.5				0.5
Detail System-Wide Requirements	Task	3		4		2			1	0.5
Create Test Cases	Task	5		1						1

- Add rate card details per role to the Team Allocation breakdown, use fixed cell formulas
- **See CUP Workbook: Sheet REFC-Allocation & Estimation [Col A-C]**

Team Allocation & Estimation Baseline

	8 hrs/day		https://ateliermdev.azurew
Rate Card $/Ho	RateCard/Day	Breakdown Element	Role
$50.00	$400.00	Analyst	Analyst

- Using fixed cell formulas, carry over the rate card details (cost per day) to the delivery process baseline and multiply the effort in days **[Col Q-AG]** by the cost per day for every task and role
- **See CUP Workbook: Sheet REFB-CUP Baseline [Col Q-AG]**

Creation Unified Process Track Map	Type	Analyst	Any Role SME	Architect	CM Spec	Depl Eng	Designer	Developer
Outline Solution Options	Task	$1,600	$400	$1,280	$0	$0	$600	$240
Assess Solutions and Estimation	Task	$800	$400	$0	$0	$0	$0	$0
Detail Use-Case Scenarios	Task	$2,000	$0	$320	$0	$0	$0	$240
Detail System-Wide Requirements	Task	$1,600	$0	$1,280	$0	$0	$600	$240
Create Test Cases	Task	$400	$0	$0	$0	$0	$0	$480

- Calculate sub-total and total estimates as required

Creation Unified Process Track Map	Type	Analyst	Any Role SME	Architect	CM Spec	Depl Eng	Designer
Sub total estimates		$22,400	$16,400	$21,360	$32,240	$11,440	$15,450
Total estimate		$290,930					

Manage and add any other non-FTE costs such as software / hardware / incremental / maintenance / taxes separately.

STEP 9: Create two main standard delivery processes: Waterfall and Iterative (Agile)

Leveraging the baseline, develop two main delivery process lifecycle models that can be altered / adapted into tailored variants to produce standard delivery processes.
It is recommended to start with one Waterfall and one Iterative lifecycle model for baselines:

- **Iterative (Agile):** If working off of the OpenUP library baseline for inspiration, you already have a pretty solid iterative baseline that may only need addition or subtraction of iterations and some redundant tasks (depending on the project profile)
- **Waterfall:** Simply start by wiping out iterations and redundant tasks to come up with a waterfall baseline.
  - **See CUP Workbook: Sheet REFD-Waterfall ➔ Notes:**
    - *Update your subtotals and totals*
    - *Effort estimates can be overridden within the variant delivery process spreadsheet, recommended to highlight any variations from baseline with justification for audit*
- You can then proceed to extend method content for delivery variants. The following variants are recommending:
  - Application Development (New Development)
  - Application Maintenance
  - External Sourcing
  - Commercial Off the Shelf
  - Infrastructure

STEP 10: Proceed to add content to assets in Eclipse Process Composer and publish variants

- Proceed to add content to the baseline process assets method content properties, including developing and associating guidance material as required.
- Learn and follow the Eclipse Process Composer (see Modules **10** and **11**) tutorials to develop process variants derived from your developed baseline
    - Use the extends functionality that EPC offers wherever possible and reduce the amount of authoring required for variants to a minimum
- Publish the baseline and variants
- Repeat appropriate elements of this described process for continuous improvement
- Iteratively document tailoring guidelines that set clear rules about what can and cannot be done to the process assets; it is recommended to create measures and metrics that relate to variances from the baseline as to encourage limiting tailoring and encourage use of standard delivery variants defined by the organization
- Proceed to Module **12** for ideas and suggestions on developing measures and metrics

10: Eclipse Process Composer Quick Setup Guide

Note: Modules **10** & **11** cite or reference the following sources ➔ [13] [14] [15] [16]

Downloading Eclipse Process Composer

https://www.eclipse.org/epf/downloads/tool/epf1.5.0_downloads.php

Requirements:

- Microsoft Windows XP Sp3 through to Windows 10
- Java Runtime Environment x86 (Win 32) ➔ 1.5 through 1.8
    - http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html

Direct download link ➔here

Windows Installation Instructions

- Expand the downloaded zip file
- Change to epf-composer folder and start the program epf.exe

Download the EPF Libraries which include OpenUP, Scrum & XP

https://www.eclipse.org/epf/downloads/praclib/praclib_downloads.php

http://www.eclipse.org/downloads/download.php?file=/technology/epf/PracticesLibrary/library/epf_practices_library_1.5.1.5_20121212.zip

Library Installation Instructions:

- Unzip the downloaded zip file
- Open EPF Composer, menu File➔Open, then point to the folder where you unzipped the library (where the library.xmi file is)

Getting Started and EPF Tutorial Documentation

https://www.eclipse.org/epf/general/getting_started.php

- Key quick manuals and getting started tutorial documents for using EPF
  - Installation, Introduction, Tutorial and Manual
  - Introduction to OpenUP: https://www.eclipse.org/epf/general/OpenUP.pdf
  - Customization scenarios with EPF Composer and Open UP:
    - https://www.eclipse.org/epf/general/EPFComposer_OpenUP.zip

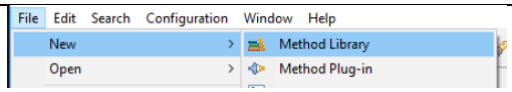
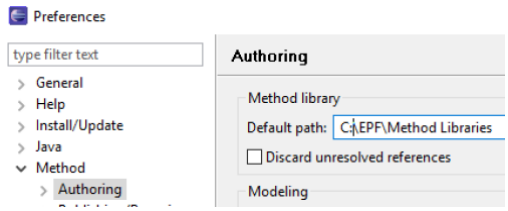
11: Author process assets with Eclipse Process Composer

Start Eclipse Process Composer (EPC)

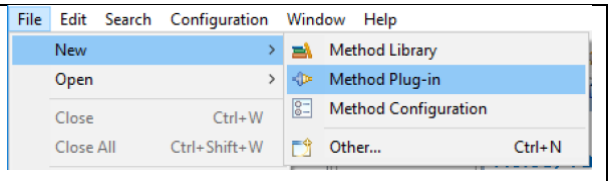
Important Note: Save your work often. ***Additional tips***:

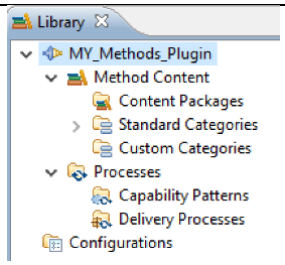
- Start with content wireframes ➔ At first, ignore the EPC bells and whistles
- Focus on developing the method content placeholder(s) and fill out the property details at a later time
- Produce/publish a wireframe delivery process that demonstrates the interaction and relationships between process assets
- By not including details, i.e. by way definitions, your audience will focus on the overall delivery process and relationships between assets (vs. intricate details), thus accelerating consensus or agreement on the high-level standard
- Process asset **details** and **guidelines** can be added subsequently as part of the continuous improvement process

Create a Method Library

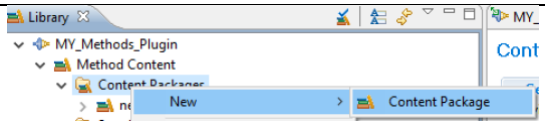
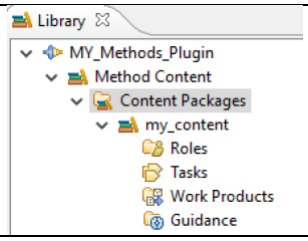
Select File->New Method Library • Provide or browse to a folder where the method content library (process assets) will be stored	
Select the Window Menu -> Preferences -> Method -> Authoring • Set the Default path for where your method libraries will be placed Select the Window Menu -> Preferences -> Method -> Publishing / Browsing • Set the Default path for published content	

Create a Method Plugin

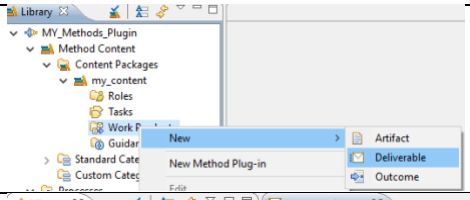
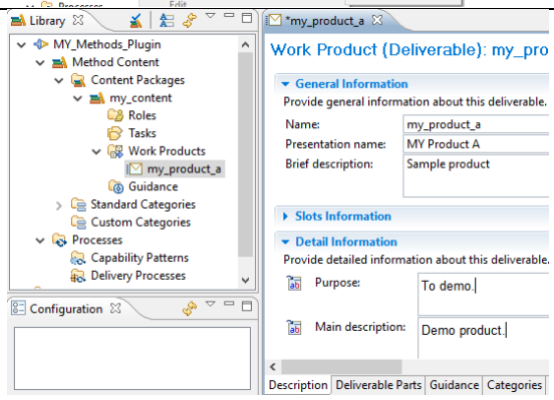
Select Window-> Open Perspective -> Authoring Select File->New Method Plugin • Provide a Name & Brief description	

Expand all the method plugin subsections in the Library window.	

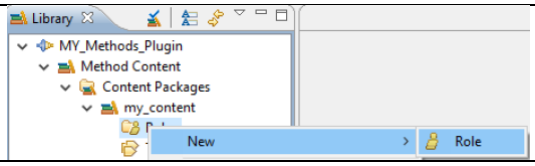
Create a Content Package

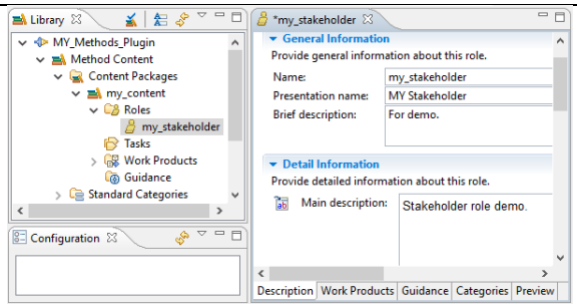
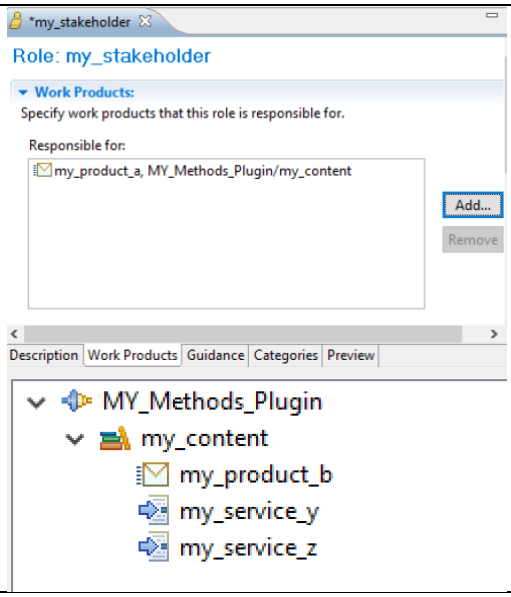
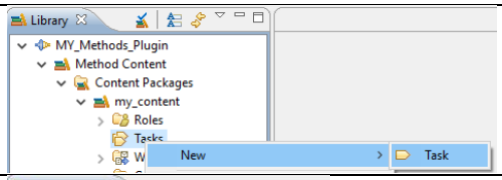
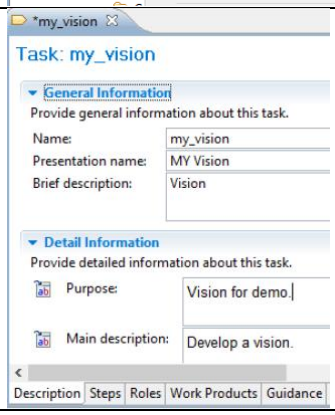
Right click and select Method Content -> Content Packages -> Content Package • Provide a Name, Presentation name and Brief description	
Expand your newly created content package in the Library window	

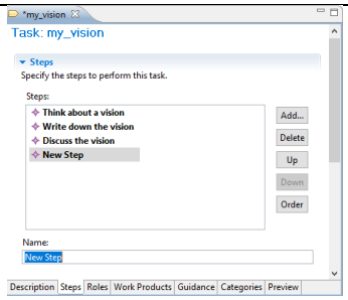
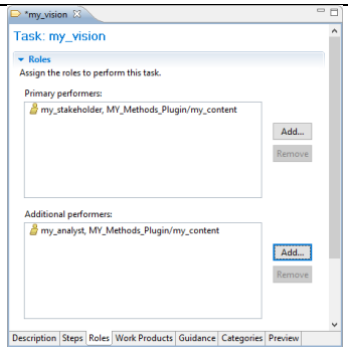
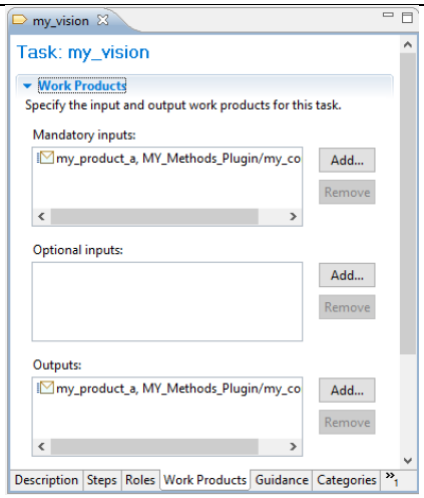
Create Deliverables

Right click and select Work Products->New->Deliverable	
In the Description section (tab), provide general and detailed information about the asset • General Info: Name, Presentation name, Brief Description • Detail Information: Purpose, Main Description **Repeat**: For all Deliverables	

Create Roles

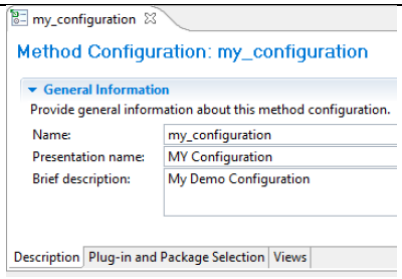
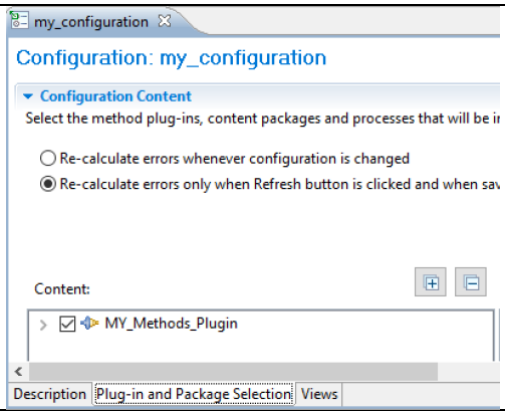
Right click and select Roles->New->Role	

In the Description section (tab), provide general and detailed information about the asset • General Info: Name, Presentation name, Brief Description • Detail Information: Main Description	
• Select the Work Products section (tab), click add beside the Responsible for section and select work product(s) the role is responsible for. Note: Multiple work products can be added. **Repeat**: For all Roles	
Create Tasks	
Right click and select Tasks->New->Task	
In the Description section (tab), provide general and detailed information about the asset • General Info: Name, Presentation name, Brief Description • Detail Information: Purpose, Main Description	

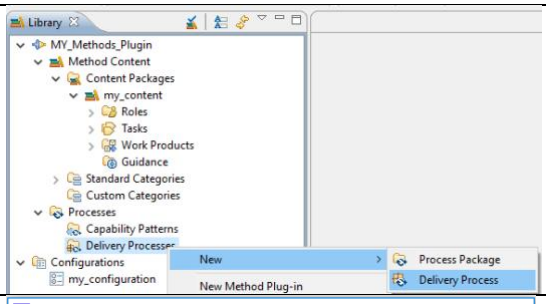
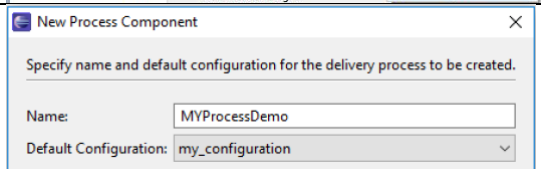
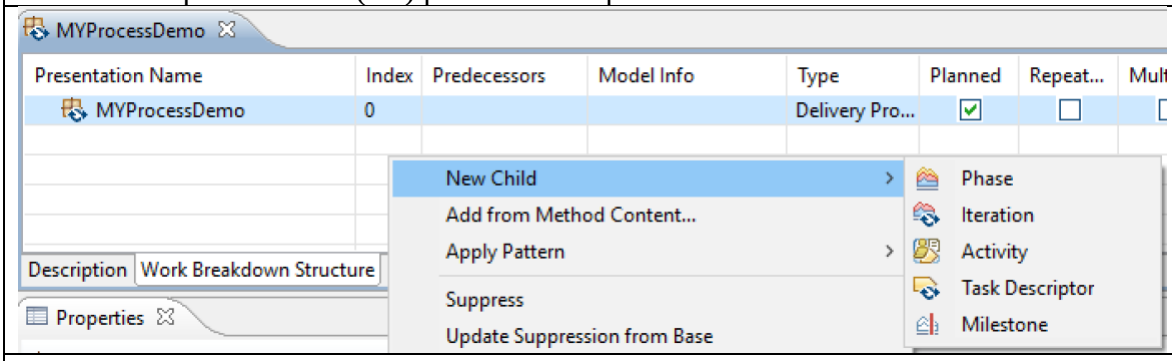
Select the Steps (tab), click add beside the Steps section and add the task steps. • Note: A detailed description of each step can be provided.	
Select the Roles section (tab) • Primary performers: Click add beside the Primary performers section and associate the role(s) that are considered accountable or primarily responsible for the task. It is highly recommended for only one role to be considered a primary performer. • Additional performers: Click add beside the Additional performers section and associate any additional role(s) involved in performing the tasks, whether responsible, contributors etc.	
Select the Work Products section (tab) • Mandatory inputs: Click add beside the Mandatory inputs section and associate the work products(s) that are mandatory or required to complete the tasks. ○ *Repeat* for Optional inputs. • Outputs: Click add beside the Outputs section and associate the work product(s) that are an outcome of performing the task.	

Create a Method Configuration

Right click and select Configurations->New->Method Configuration	

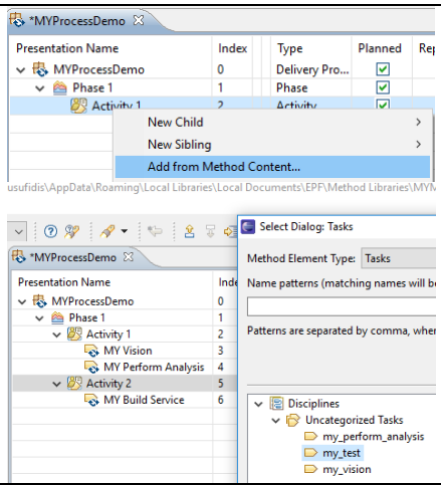
In the Description section (tab), provide general and detailed information about the asset • General Info: Name, Presentation name, Brief Description	
Select the Plug-in and Package Selection section (tab) and in the Content section select the Methods Plugin created earlier. All sub-content should be checked.	

Create a Delivery Process

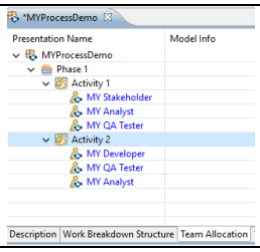
Right click and select Processes->Delivery Processes->New->Delivery Process	
Provide a name and select the default configuration created earlier from the drop down	
In the Description section (tab) provide description details.	
Select the Work Breakdown Structure section (tab) and right click within the delivery process window to begin building the delivery process by selecting New Child and building the required process components.	

- **Phase, Iteration and Activity** details can be provided via the Properties window and is similar to how method content is built.
    - Keep in mind, if **Waterfall**, you'll have **no** iterations: simply,
        - Phases ➔ Activities ➔ Tasks
- A **task descriptor** is essentially a delivery process specific task that is not part of your content baseline, so a task that is unique to the delivery process
- When **adding tasks**, it is highly recommended, wherever possible, to add tasks from **Add Method Content**, which is where you can access the developed Method Content, rather than adding unique Task Descriptors
    - The value in doing so is that the relationships established while building the method content are carried over into the delivery process; this can be continuously evaluated as you're progressing while building the work breakdown structure for your delivery process by selecting the Team Allocation and Work Products tabs

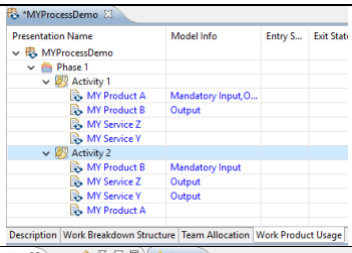
*The value in doing so is that the relationships established while building the method content are carried over into the delivery process; this can be continuously evaluated as you're progressing while building the work breakdown structure for your delivery process by selecting the **Team Allocation** and **Work Products** tabs, see below for an example*



Team Allocation: View

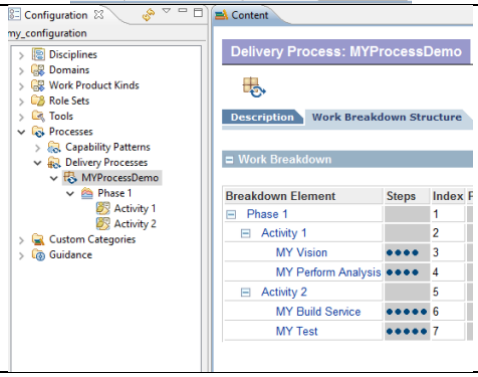


Work Products: View

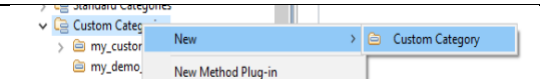
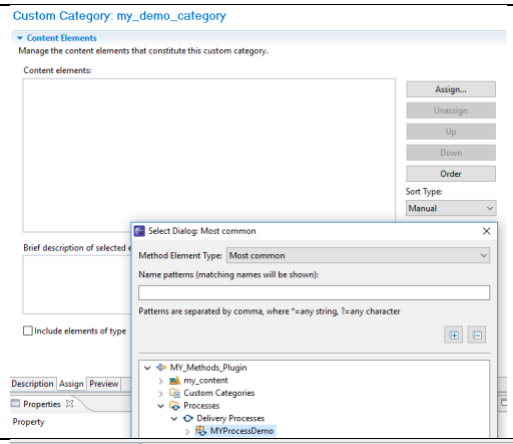
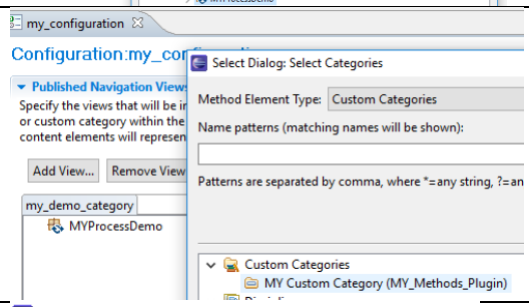
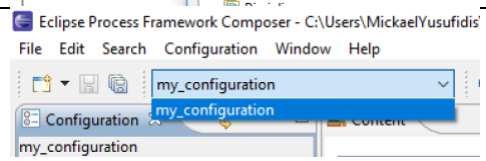
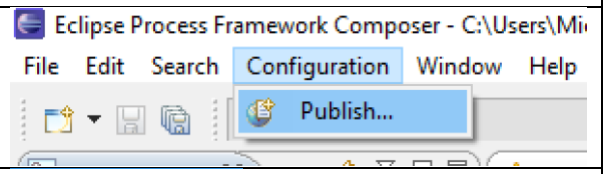
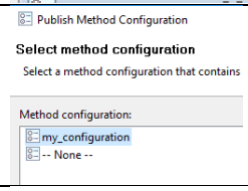
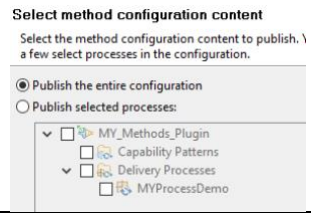


To Preview the Delivery Process:

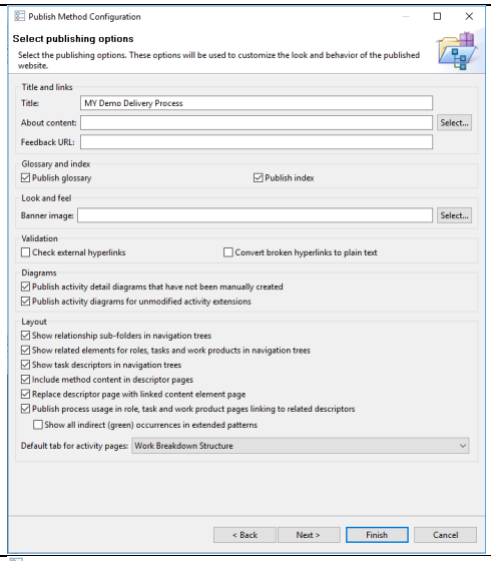
- Select Window-> Open Perspective -> Browsing
- From the configuration window expand Processes and Delivery Processes. Select the Delivery Process to preview.



Publish the Delivery Process

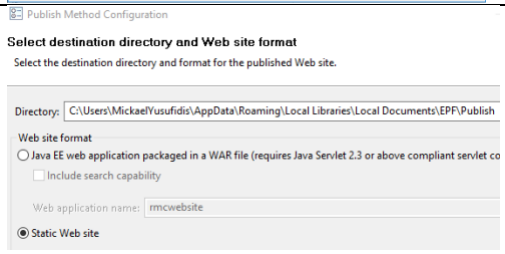
From the Authoring: Library view, select Method_Plugin->Method Content->Custom Categories->New->Custom Category • Create a custom category	
Assign the created Delivery Process to the custom category.	
From the Authoring: Library view, open the configuration that was created in the Create Method Configuration step(s). • Click on Add View and select the Custom Category that was created	
Select Window-> Open Perspective -> Browsing. Select the configuration that will be published from the configuration drop down.	
Select Configuration -> Publish	
Select the Method Configuration and click on Next (Not Finish).	
Select Publish the entire configuration and click on Next (Not Finish).	

Provide a title and make the following configuration selections. Click on Next (Not Finish)

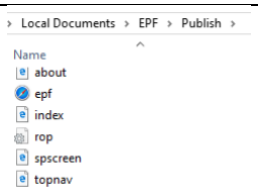


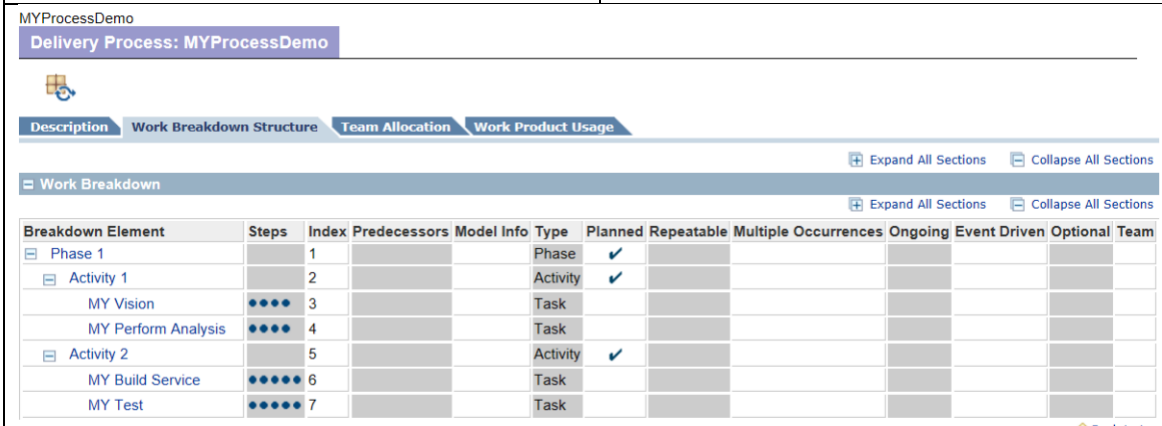
Select Static Web site and if required adjust the Directory from what was originally set in Preferences.

- Click Finish and OK to any warnings if publishing to a directory where content already exists (previous publish)



The content found within the defined publish directory can be accessed locally via the index.htm file or copied into any web services directory and accessed via the index.htm file.





12: Measures + Metrics: Improvise, improvise, improvise

Here are some abstract ideas as a guide for discovering things to measure and develop fit for purpose metrics on your own. All you have to do is look at the process assets that you have developed, think of how they can change depending on variants and you probably come up with some interesting metric/measures, which you can also combine with the one metric to rule them all to get a quick roll up dashboard running, i.e.:

- **Measuring the project's health, progression and variance from the baseline**
  - The delivery of a business case and approval to proceed to Execution is considered the project's point of singularity or ground zero ➔ 0
    - All activities prior (Ideation, Initiation and Planning) are considered unprogressive and measured as -1
    - All activities after business case approval are considered progressive (closer to product/service realization) and measured as +1
    - This type of metric encourages limiting effort spent in initiation and planning and progressively working towards delivery or realization of the deliverable

- **Tailoring results in varying from the fixed work breakdown structure and estimates, given the baseline could represent fixed project delivery costs**
  - Any project activities/tasks that are modified or added to the baseline as to add effort, project complexity and cost from the fixed baseline are considered unprogressive and are measured as -1
  - Any project activities that are modified as to reduce effort, project complexity and cost from the fixed baseline are considered progressive and measure's as +1

The opportunities are endless, afford yourself some creativity and improvise. Most importantly, you have the ability to be as granular (detailed) or as high level as you want. Try to find the middle of the road and keep the metrics simple and relatable so you can easily perform roll up reporting. Lastly, go find the metrics specialist in your organization and have a discussion with them. Ask them how they feel about the one metric to rule them all and if left up to their imagination, what they could do with it.

Works Cited

[1] Eclipse Process Framework: OpenUP, "OpenUP: EPF Wiki Eclipse," Eclipse, 2018. [Online]. Available: http://epf.eclipse.org/wikis/openup/.

[2] M. Yusufidis, "OpenUP: Creation Unified Process," 2016. [Online]. Available: https://ateliermdev.azurewebsites.net.

[3] M. P. Ginsberg and L. H. Quinn, "Process Tailoring and the Software Capability Maturity Model," Software Engineering Institute: Carnegie Mellon University, Pittsburgh, PA, U.S.A, 1994.

[4] M. Yusufidis, *Process Tailoring Overview Figure (Simplified & Updated)*, 2018.

[5] Oxford Dictionaries, "Oxford Dictionaries," Oxford Dictionaries, [Online]. Available: https://en.oxforddictionaries.com/definition/allusion.

[6] Merriam-Webster, "Merriam-Webster: Definition of Allusion," Merriam-Webster, [Online]. Available: https://www.merriam-webster.com/dictionary/allusion?src=search-dict-box.

[7] W. Petri, The Universal Version Bible - The Prophetic Scripture and Appendixes, Beacon Ministries Publication, 2016.

[8] Wikipedia, "Wikipedia: Magic Realism," Wikipedia, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Magic_realism.

[9] Wikipedia, "Wikipedia: Improvisational Theatre," Wikipedia, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Improvisational_theatre.

[10] J. Rae-Dupre, "Can Executives Learn to Ignore the Script," New York Times, 2 March 2008. [Online]. Available: https://www.nytimes.com/2008/03/02/business/02unbox.html.

[11] P. Smith, "10 Fundamentals Principles of Improv comedy," Jesterz Improv Comedy, 30 January 2011. [Online]. Available: https://www.jesterzimprov.com/fundamental-improv-principles/.

[12] M. Bennett, "4 Rules of Improv and How They Relate to Customer Support," Zapier, [Online]. Available: https://zapier.com/learn/customer-support/improv-customer-support/.

[13] Eclipse Process Framework, "Eclipse Process Composer 1.5.X Downloads," Eclipse, 2016. [Online]. Available: https://www.eclipse.org/epf/downloads/tool/epf1.5.0_downloads.php.

[14] Oracle, "Java SE Runtime Environment 8 Downloads," Oracle, 2018. [Online]. Available: http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html.

[15] Eclipse Process Framework, "EPF Practices Library Downloads," Eclipse, 2016. [Online]. Available: https://www.eclipse.org/epf/downloads/praclib/praclib_downloads.php.

[16] Eclipse Process Framework, "Eclipse Process Framework: Getting Started," Eclipse, 2016. [Online]. Available: https://www.eclipse.org/epf/general/getting_started.php.