

A photograph of a person's hands working on a laptop and a tablet. The person is wearing a dark jacket and glasses. The laptop is open on a wooden desk, and the tablet is held in the foreground, displaying a bar chart. The background is slightly blurred, showing a professional office environment. A blue semi-transparent overlay covers the left side of the image, containing the title and author information.

# ARCHIMATE, TOGAF AND OGC'S MSP

Written by:  
Peter Harrad

# INTRODUCTION

---



One thing this world is not short of is frameworks and guidelines. Just considering the area in which Orbus commonly plays, we have TOGAF (and DODAF and MODAF and a number of others AFs), we have COBIT, we have IT4IT, we have... well, the list goes on and on. In this forest of publications it becomes easy to ignore some of the less well known ones, which is a pity, because at the very least they often have insights or ideas of real value.

Last year I had cause to get involved with a public sector project where I came into contact with the Managing Successful Programmes methodology from the UK Government's Axelos intellectual property venture – the same people that are behind ITIL and PRINCE. Managing Successful Programmes, or MSP, is the official UK government methodology for program management; UK government departments are required to use it, and local governments are strongly encouraged to. In this ebook I'm going to take a look at the key ideas of MSP and how they compare or dovetail with the TOGAF and Archimate standards.

Throughout the eBook, I'll be using the example of a local government authority who are implementing a transformation programme based on better management of data across disparate authority systems.





# AN OVERVIEW OF MANAGING SUCCESSFUL PROGRAMMES

---

The Managing Successful Programmes is designed to be a structure for managing transformation of some kind within an organization, defined as a set of individual projects that are grouped into a program. It very explicitly focuses on how to govern the program in order to ensure not only successful delivery, but delivery that achieves the end goals. To this end, where it stands out from other frameworks that I have seen is a focus on ongoing monitoring of benefits.

To assist in achieving its goals, MSP defines nine core Governance Themes;

## **Organization**

Obviously, any programme needs an organizational structure with reporting lines to enable proper governance of the programme. An interesting aspect of MSP is an insistence on lean reporting lines; that is, MSP requires that any given role across the whole programme must have a maximum of two and if possible only one reporting line; with a second reporting line acceptable only where some level of technical management is required.

## **Vision**

MSP requires that the first stage of the programme is defined in a Vision Statement, that outlines the high-level description of the end state after the programme completes. Depending on the situation, it might be defined by a core internal team or be the result of a more collaborate, group effort.

## **Leadership & Stakeholder Engagement**

It would be strange to have a framework for transformation that did not include stakeholder analysis as a key activity. Yet there is an important subtlety in the MSP approach to stakeholder management, which is that the concerns for each stakeholder must be drawn out and explicitly mapped. This is necessary because they form inputs into the later themes of benefits realization management and quality management – calling out the stakeholder satisfaction aspect.

## **Benefits Realization Management**

One of the most distinctive features of MSP is drawn from the keen insistence on satisfying stakeholders – Benefits Realization Management. Essentially, this theme states that every programme should have a map of the expected beneficial outcomes, known as benefits (and the detrimental ones, known as dis-benefits). Each benefit should be measurable; where an identified benefit is not, it should be mapped to a resulting benefit that can be. This becomes important for the business case. At the same time, benefits should be mapped to stakeholder concerns; some benefits may only satisfy certain stakeholder concerns, while others might be more universal.

# AN OVERVIEW OF MANAGING SUCCESSFUL PROGRAMMES

## Blueprint Design & Delivery

The Vision Statement defined at the start of the programme is used for an initial definition and evaluation of the programme; but it is not enough for formal definition. The programme activities and outputs are fleshed out in a document known as the programme blueprint. An MSP blueprint considers change from four aspects, known collectively as POTI:

- People Changes
- Organization Changes
- Technology Changes (including physical assets such as physical premises)
- Information Changes

## Planning & Control

An MSP programme is divided up into tranches; each tranche being a incremental delivery of functionality. This ensures that even if a programme is aborted midway through (something that can happen in any organization, but which is particularly a risk in government), the programme will still have delivered some form of partial value. This in turn means that an important aspect of MSP is that a post-tranche review is performed as each tranche is delivered.

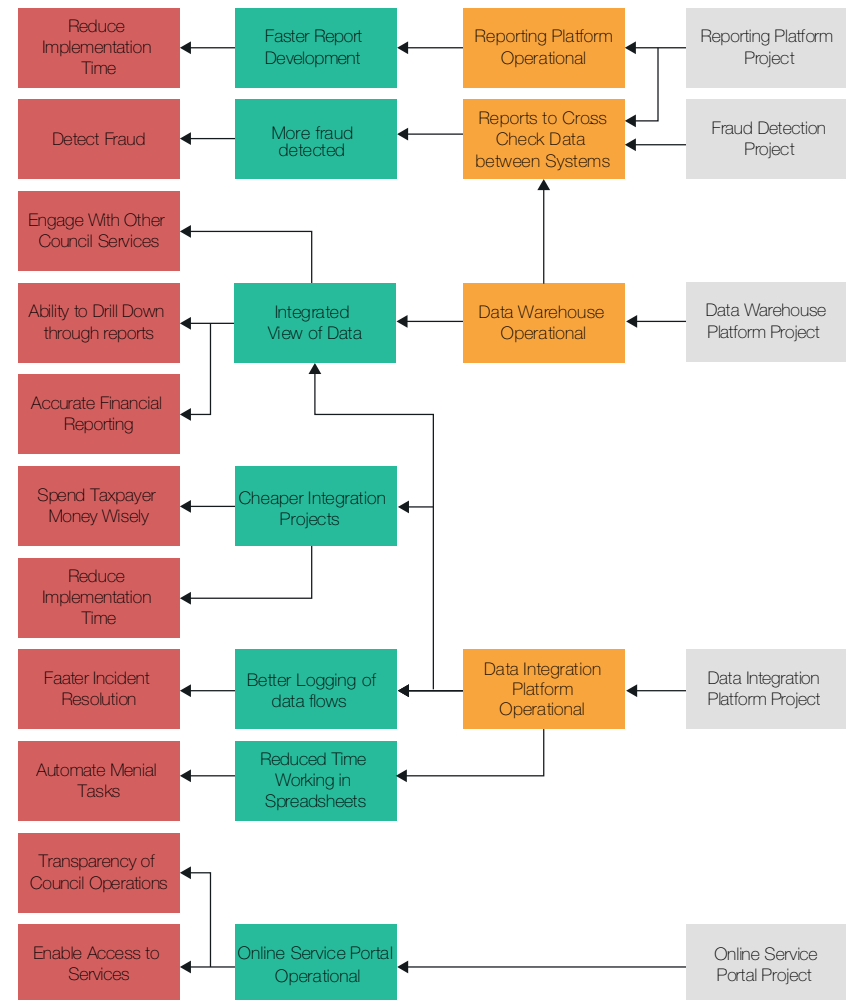


Figure 1: A Sample Benefits Map



# AN OVERVIEW OF MANAGING SUCCESSFUL PROGRAMMES

---

## **Business Case**

Again, a business case is an expected part of any programme, and hence any programme management framework needs to consider how business cases are constructed. Where MSP is interesting, is that the business case is built using the blueprint but also the benefits map discussed earlier. Since one aspect of benefits management is that each end benefit (or disbenefit) in the chain is required to be measurable, they are explicitly designed to feed into the business case.

## **Risk Management and Issue Resolution**

Risk Management and Issue Resolution are also expected components of a programme management framework, and MSP devotes some time to consideration of risk and issue management.

## **Quality Management**

It should not be controversial that a framework for successful programme management would have something to say on the subject of quality management – and some of MSP's guidelines on this subject are unsurprising. However, here as elsewhere the focus on stakeholder management stands out. The MSP approach to quality management recommends taking the stakeholder-concern mapping from the stakeholder analysis, and mapping these concerns to CSFs (Critical Success Factors) that programme management then uses to derive KPIs.

Parallel with these nine governance themes, the Managing Successful Programmes framework also defines six core processes;

### **Identifying a Programme:**

The process of the initial definition of the Vision Statement for the programme

### **Defining a Programme:**

The process by which the Benefits Realization Plan, Blueprint and Business Case are defined

### **Managing the Tranches:**

The process by which individual tranches are officially started, monitored and closed off as they complete

### **Delivering the Capability:**

The process by which each individual project is started, operated and closed as they complete

### **Realizing the Benefits:**

The process by which each change required by the programme is transitioned into operation

### **Closing a Programme:**

The process of closing the overall programme once all tranches and projects have completed, including a post mortem review


# TOGAF & MSP

---

It is likely that someone familiar with TOGAF will have already recognized similarities between some of the aspects of MSP and TOGAF. However, there are aspects around value delivery and governance that a TOGAF programme can usefully draw from MSP. Let us consider each aspect of MSP that touches onto TOGAF, in turn.

## Blueprint Design

The MSP Blueprint arguably maps to a combination of the TOGAF Statement of Architecture work and the Architecture Requirements Specification; yet what they both lack is the formal approach of considering changes from specific perspectives such as the POTI model.



*"This approach has benefits to TOGAF, in that it keeps a focus on delivery of value; at the same time, architectural transformation becomes an easier 'sell' if there are regular 'go/no-go' checkpoints."*

## Benefits Realization Management

Benefits Realization Management is probably the single part of MSP that has the most to offer to TOGAF. While certain artefacts and deliverables in TOGAF, such as the architecture requirements specification and the driver-objective-goal catalog, there is no formal prescription to specifically link individual outcomes to measurable benefits which satisfy individual stakeholder concerns.

## Tranches

The concept of tranches (an incremental delivery of functionality) has clear analogues in the TOGAF concept of transition architectures. Both represent intermediate stages between the start state and the end state. However, tranches differ in two ways. First of all, the concept is more general, in keeping with the need for MSP to address every possible kind of programme. More importantly, MSP adds the concept of review after each tranche, something that is perhaps implicit in TOGAF, but not called out explicitly. That is, after each incremental delivery the programme reviews the business case for continuing. This approach has benefits to TOGAF, in that it keeps a focus on delivery of value; at the same time, architectural transformation becomes an easier 'sell' if there are regular 'go/no-go' checkpoints.

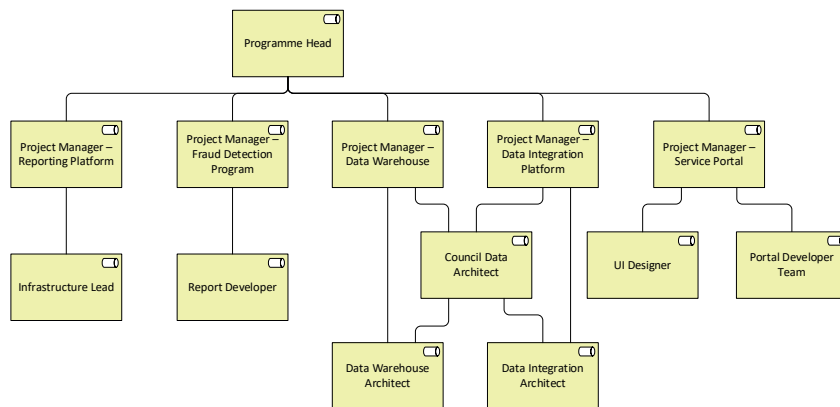
# ARCHIMATE & MSP

There are several aspects of MSP that can benefit from direct modeling. In this ebook I'll be using the Archimate modeling language, but other languages could also be effective.

## Reporting Lines

The first area where Archimate could assist MSP is in modeling the reporting lines. Defining clear reporting lines for each role in each project in the programme is a key concern when defining the programme structure, so the ability to map roles to each other becomes important. In Archimate, we can accomplish this by mapping the roles in the structure to Business Roles, which Archimate defines as 'the responsibility for performing specific behavior.'

Taking it further, each person fulfilling a role in a given project could be mapped as a Business Actor and assigned to the Business Role that they perform in that project.



## Stakeholder Management

The next area where Archimate seems to offer benefits to an MSP programme is in the Stakeholder Management aspect. As described earlier, satisfying stakeholder concerns by delivering benefits is a key foundation of MSP. Using Archimate, we can map Stakeholders as the Stakeholder object (strangely enough), who have Drivers which represent their concerns.

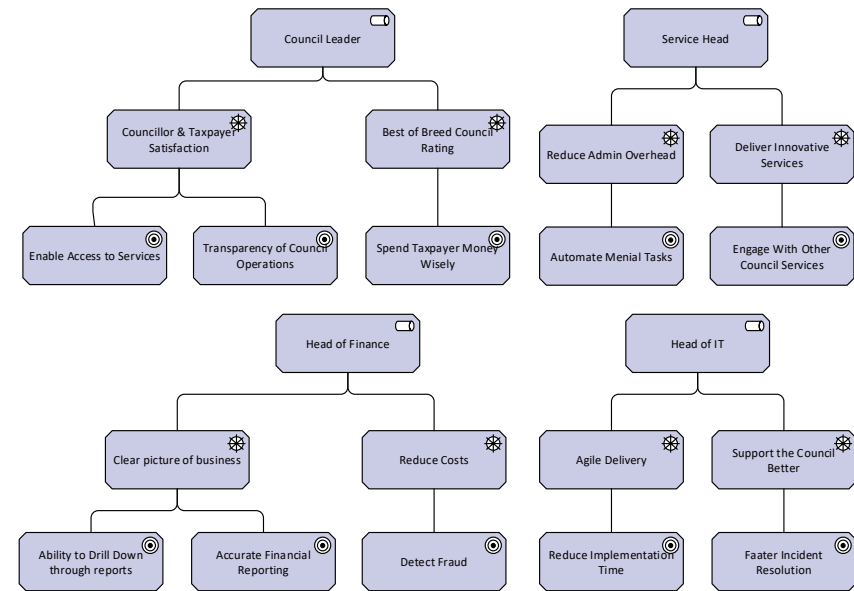


Figure 2: A sample Stakeholder Map in Archimate with concerns mapped

# ARCHIMATE & MSP

## Benefits Realization Management

The MSP approach of mapping outcomes to benefits has clear analogues to the Archimate concept of a Motivation Model. Since Stakeholders (mapped as Stakeholder objects) and their concerns (mapped as Archimate Drivers) already exist as outcomes of the stakeholder Management exercise, they are readily available to us to map them to outcomes and goals. Specifically, we can map the project outputs as Archimate Outcomes, which we then map to Archimate Goals which represent the programme benefits. Per Archimate, a Goal can be qualitative (i.e. non-measurable) or quantitative (i.e. measurable). These benefits are then mapped to the Drivers that came from the stakeholder mapping exercise.

## Quality Management

Again, since the MSP approach to quality management depends on performing mappings, Archimate seems to offer a solid platform for accomplishing this activity. As described earlier, the stakeholder-concern mapping can readily be performed in Archimate. Thus, these concerns become immediately available for mapping to Critical Success Factors and hence to Key Performance Indicators.

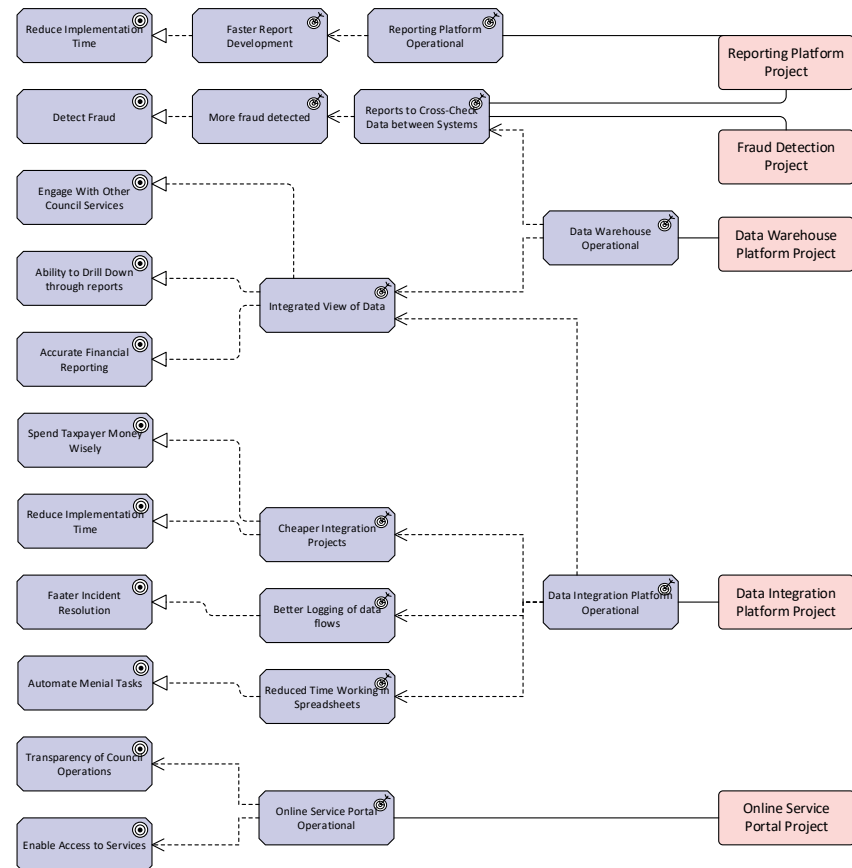


Figure 3: A Sample Benefits Map in ArchiMate



# ARCHIMATE & MSP

## Tranche Management

The next area where Archimate modeling could assist MSP is in tranche management; specifically, in mapping the projects that make up the programme to the tranches. A tranche in MSP has a clear mapping to a Plateau in Archimate. Likewise, a project in MSP can easily be modeled as a Work Package in Archimate; while the deliverables of the project map to Archimate Deliverable objects. Thus each Deliverable for a given Work Package (project) can be clearly mapped to a Plateau (tranche).

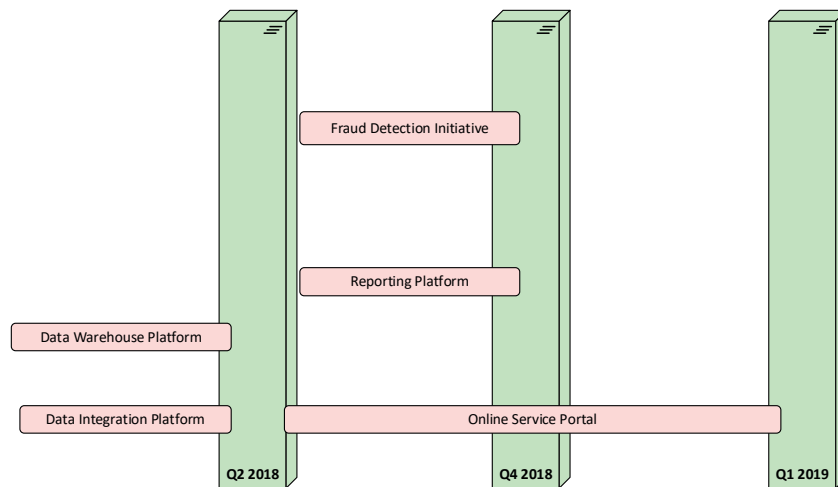


Figure 4: A sample Tranche Map in Archimate



*In each process, the steps within the process can be clearly mapped to the role that performs them.*

## Process Mapping

The last area where Archimate can help in an MSP programme would be to map the processes that exist; programme initiation, post-tranche review and so on. In each process, the steps within the process can be clearly mapped to the role that performs them. Of course, it is possible to map this in BPMN as well – an important question for this is whether the modeling tool being used can support mapping BPMN entities such as lanes and processes to Archimate entities such as Business Roles and Business Processes.

## CONCLUSION

---

It turns out that there are numerous similarities between TOGAF and the Managing Successful Programmes framework. This is unsurprising given that they both deal with how to accomplish transformation, but TOGAF is focused on enterprise architecture transformations, MSP is more generalist and so has a greater focus on governance and stakeholder management. This means in turn that there are a few techniques that TOGAF could benefit from borrowing from MSP.

At the same time, it turns out that Archimate modeling would be highly effective in implementing these techniques, once borrowed from MSP.

Note: For more information about MSP, go to <https://www.axelos.com/best-practice-solutions/msp>





© Copyright 2018 Orbus Software. All rights reserved.

No part of this publication may be reproduced, resold, stored in a retrieval system, or distributed in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

Such requests for permission or any other comments relating to the material contained in this document may be submitted to: [marketing@orbussoftware.com](mailto:marketing@orbussoftware.com)

Orbus Software  
Portland House, Bressenden Pl, Westminster, London SW1E 5BH

+44 (0) 20 3824 2907  
[enquiries@orbussoftware.com](mailto:enquiries@orbussoftware.com)  
[www.orbussoftware.com](http://www.orbussoftware.com)