AGLPM1 - Unit 3 - Activity 2: OBSERVE

2nd Agile Leadership Skills: Harness Change

This document is an excerpt from the book:
"Agile Project Management for Government "
Authored by Brian Werham
Published by Maitland & Strong
Reproduced with permission under license

PMCAMPUS.com / Mokanova Inc

.

.

Agile Leadership Behavior Two: Harness Change to your Advantage

> *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*
>
> Agile Manifesto Principle Two

One of the great things about agile project management is that it creates a natural restlessness with regard to the objectives of a project. By assuming that change is inevitable, it tries to seek out that change at an early stage, to facilitate it quickly, and with minimum resistance.

Many projects produce business cases that are so detailed that they become an albatross around the project manager's neck. Changing even minor details of a project's objectives can be fraught with difficulty when faced with the need to keep an overcomplicated business case in line with latest thinking.

The agile approach encourages creativity in thinking in governments. It encourages small, frequent changes of direction rather than only making changes when forced to – which is late, often difficult, and always painful. While, up to now, many technology development projects have adopted a waterfall approach, thus encouraging introversion, we will see agile approaches in the future encouraging extroversion. Waterfall projects rely on commenting on change control documents at a distance, rather than meeting stakeholder face-to-face and collaborating.

Both the US and the UK Governments face challenges in implementing agile project management so as to harness change in this manner. The US Congress must consider what it can do to facilitate an agile approach without tying down government with prescriptive legislation. Both Governments must review their rulebooks and make them consistent with the 9 Agile Leadership Behaviors.

Seek Out and Harness Change

People working on and those affected by government technology development projects react to potential change in three different ways (see **Figure 1**):

- ♦ Resisting change
- ♦ Seeking change
- ♦ Harnessing Change

There is a risk that projects can burn up sums of money before any benefits are in sight. A waterfall approach can result in a series of feasibility studies and reports, a business case and requirements analysis before any practical development takes place.

A natural inclination of some involved in government projects is to try to manage risk by creating spurious convincing detail in a business case. In the UK, for example, Government departments are required to follow the Treasury Green Book regulations. Over the years these regulations have been widened and deepened. The Green Book guidance was last updated substantially in 2003 and includes recommendations on writing business cases for projects. For example, it gives reasonable advice to recognize and value any 'optimism bias' in proposals, and ensure that judgmental estimates for risks are quantified and shown separately in the calculations.[i] The guidance covers projects that create *intangible* assets such as the technical intellectual property created by software development projects.[ii] It is a readable and terse document compared with the OMB Capital Programming Guide, but it still has a tendency to create pressure for voluminous documentation. This often causes early commitment to plans of action based on prematurely agreed large-scale solutions before they have been proven to work on a small-scale.

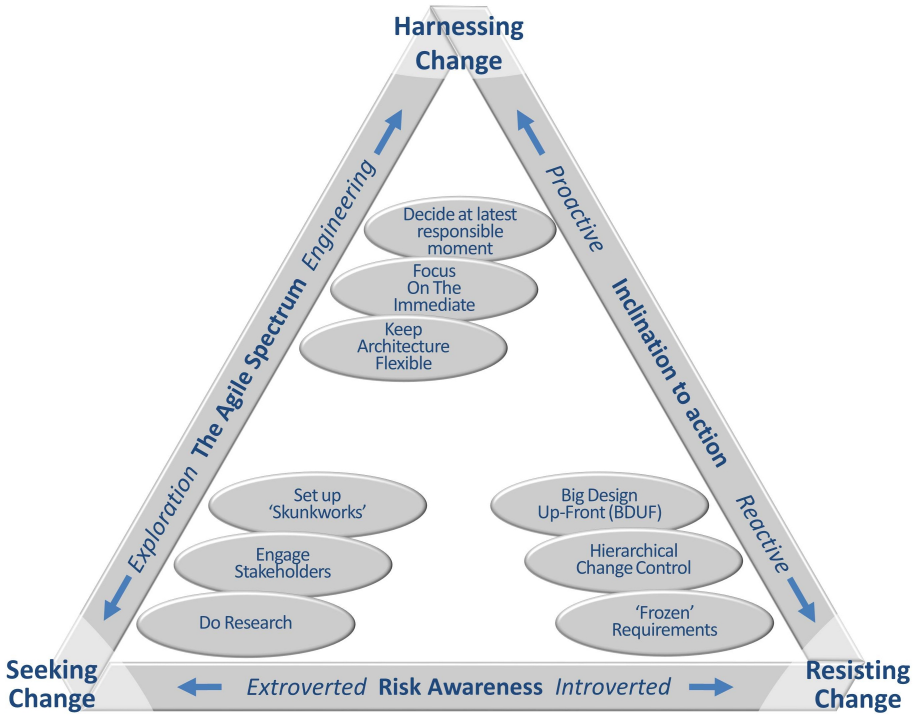

Figure 1: Inertia and Resistance to Changes in Project Direction

The NAO has criticized the regulations for encouraging *spurious precision* used as the basis for major decisions. Often, they warn, the justifications are "pseudo-scientific mumbo jumbo".[iii] The NAO gives one example where a hospital project for £746.1m was approved on the basis of such a calculation showing just a £0.1m saving.

The UK Parliamentary Accounts Committee (PAC) is a powerful bi-partisan body of Members of Parliament chaired by a member of the opposition party. The PAC has a powerful role of scrutiny into Economy, Efficiency and Effectiveness of the UK Civil Service. These three "Es" are their definition of *value for money* (VFM). The committee's role is not to find fault with policy, but to examine the performance of the Permanent Secretary of each government department. They are supported in this investigation by the work of the NAO, which is not part of the regular civil service, but is independent, and directly funded by parliament. The usual chain of events is that the NAO will carry out VFM audits, with powers of inspection and audit with not only government staff but staff in the private sector supply chain. The NAO publishes a report on each audit stating whether VFM has been achieved. The PAC then calls witnesses to parliament to be questioned. These entertaining events are shown live on webcasts and on UK digital TV.

Both the NAO and the PAC have been critical of the justification of large projects based on artificial business cases:

> "Spurious precision is unproductive … public authorities need to recognize degrees of uncertainty … slight adjustments to the calculations, well within the range of error … (are often made to) ensure cost(s) appear marginally cheaper." [iv]

Often the time needed to create this detail creates risk because it closes down the investigation and discussion of other options that may later be seen as superior, and need to be revived (or at least kept in reserve as a 'Plan B' in case the initial project design flounders).

Once this attitude towards the need for detail has started, there is a natural inertia to change. All the careful (i.e. over-detailed) planning that has gone on upfront may start to unravel if any of the base assumptions change. A detailed business case often is fragile, rather than robust. As work proceeds defenses are erected to defend the embedded assumptions. The first defense is to create an even more detailed set of requirements to support the assumptions in the business case. The objective is to *baseline* the documented requirements, and then *freeze* that baseline, not to test it practically and improve upon it. Perversely, this, in conjunction with obscure notation and voluminous documentation, may encourage requirements analysis to be done in isolation from stakeholders.

Bureaucratic Approaches versus Creative Solutions

Orlikowski and Beath have noted the problems arising from this tendency to prematurely *baseline* and *freeze* requirements. They cite an example from some 'best practice' guidance materials they reviewed:

> "The data administrator, with the help of systems analysts, fed the user views of data into a data modeling process and into the (CASE tool). Appropriate printouts of the models, from the encyclopedia, are given to the user committee to check." [v]

This bureaucratic approach had two self-serving payoffs for the project team, they argue: first there was a release from the obligation to fulfill the stakeholders' real needs, and second it allowed the project team to drive the development in a predetermined direction.

However, as a project progresses there are several threats to the apparent stability of the business case and its supporting requirements. First, outside circumstances may change. Second, hidden requirements may inconveniently start to appear. And third, as development begins, it may become apparent that aspects of the chosen solution design are not workable. These potential changes threaten the investment (both financial and psychological) in the existing Big Design Up-Front, and often defensive measures are placed in the way of acceptance that change is needed, and needed fast.

A heavy handed *change control* process may be used that fosters inertia rather than reacting swiftly to necessary changes.

There are often three stages to resistance. First, when a potential change is recognized, an *impact analysis* will need to be conducted to work out what the impact of the change will be. A lot of documentation may need updating if even a small change is made. Second, this impact analysis will be long and complicated due to the large size and number of designs that have already been created, and the complex interdependencies between these designs. So before the impact analysis can start, somebody (usually a change control committee or a project board) has to authorize the effort needed to carry it out. Third, in a hierarchical control mechanism, any effort needed to be spent carrying out that impact analysis will need a documented justification, which will need time for discussion and agreement. The nature of this whole mechanism is to add time and budget to the project, and to discourage and slow down the necessary process of change and adjustment.

Ignoring Risks

Another strategy which may be adapted to those projects that are heavily committed to one course of action, come what may, is to avoid engagement with stakeholders. Several symptoms may be exhibited:

- An introverted, technical focus that makes the objectives difficult to understand
- Large, unwieldy specifications
- Inadequate stakeholder liaison
- Lack of real stakeholder buy-in
- Hidden requirements that will have to be incorporated later at a large cost and risk.

The Agile 'Spectrum' of Responses to Discovery and Change

Agile Leadership Behavior Two encourages a welcoming approach to clarifications that occur as detail is probed. The DSDM framework, for example, differentiates between 'Exploration' activities during an iteration of analysis, and 'Engineering' activities during build. Both can be built into an iteration of work so that 'Exploration' activities bring to the surface new, previously hidden requirements, and discover detail within existing

requirements. (Often agilists use the word *surface* as a verb, and talk about previously hidden requirements *surfacing* when they are discovered by the team.)

The DSDM framework encourages techniques such as stakeholder workshops, and recommends that facilitators of such workshops should be independent of the people present. Mock-ups of the end-solution may be brought to such workshops to facilitate discussion of options. This is especially important at the interface between what the stakeholders would like in a perfect world, and what the available technology can provide. Examples of these mock-ups may be partially working computer programs, scale-size models of buildings, or plans and diagrams. The more realistic these can be made the better.

Specification Prototyping

Partially working models may be developed for the purpose of illustrating the possibilities and restrictions of the possible technology solutions, and often these models are not intended to be robust or of the necessary scale to be implemented – these are *specification prototypes*.

These throwaway prototypes are particularly useful where the requirements are novel and not well understood. Large, detailed written requirements specifications are difficult to critique when the overall shape of desired outcomes and benefits is not understood. This can result in tedious document review sessions where the detail is picked over and corrected for detailed accuracy in logic, while the overall assumptions are still in debate. Researchers have criticized the usefulness of such review sessions when compared with the use of working system models which can demonstrate and bring to life how new processes would work. The aim of such functional prototype review sessions should not be to design the whole system, but to tease out and confirm (or otherwise) the assumptions that need verification. Such prototyping activity can typically cost about 10% of the eventual development effort, but it saves much more in expensive rework later in the software life cycle.[vi]

Setting up of Covert Skunkworks

Another strategy for getting your stakeholders involved, is to set up a *skunkworks*. This is a team room where everybody is working together in close proximity, often covertly without explicit budgets or formal approval from management. Being short of official funding, such teams are often working semi-covertly, housed in ramshackle surroundings that are none too fresh – hence the term.

The term originated from Lockheed Martin, who developed the ‘stealth’ ground attack aircraft which was nearly invisible to radar. With no red-tape to slow them down, the Lockheed skunkworks developed several prototype demonstrators within two years of being awarded the stealth research contract, and within five years a successful test

flight of the F-117 took place. A Lockheed executive explained the success:

> "The skunkworks approach demands the use of a small number of high quality individuals staffing each function. Individuals are given broad responsibility and have substantial workload. Our experience has shown that under these circumstances individual achievement is most often much higher than management's expectations." [vii]

Iterative Development

In an agile approach, once a minimum set of general requirements have been identified, a short burst of development takes place to create a partial solution. The objective is to explore the requirements and demonstrate that the technology can be made to work satisfactorily. This is done in a short timescale so as to get feedback on any problem areas as fast as possible. Therefore only part of the overall requirement is tackled in each burst of work.

Three outcomes are possible for each requirement: it may be validated, it may be found to be irrelevant or it may need refinement. Sometimes a requirement may be validated, but found to be of less importance than thought, and so any further work on it may be deferred. Development is repeated in this manner with some outputs not just being delivered for demonstration, but for operational use. Then the cycle is repeated until enough has been delivered, and it is assessed that any further work is not adding much more value.

This iterative delivery is part of the core of an agile approach, but does not define agile in its entirety – as we saw in the FBI Sentinel project, phased delivery of solutions is not sufficient in itself.

Agile terminology may vary from method to method, but the principle and the practice are similar.

In DSDM, development work is termed the *engineering activity*, and the output of each iteration is called the *emerging solution*. In Scrum the outputs are *potentially releasable increments*. Under both methods, each item developed, whether an item on a Scrum product backlog or an entry on a DSDM Prioritized Requirements List, needs a tight completion definition to ensure that the correct quality is achieved. In Scrum each feature delivered in a sprint is described as *done* when it meets a *definition of done* set by the Scrum team. The stricter the quality requirements, the more time will be needed per feature, and therefore less will be attempted in that sprint. Progress on these features will be tracked in the *sprint backlog*. It is not specified in the Scrum manual as to when this definition is agreed, so it is important to agree how and when these definitions of done will take place. *Definitions of done* should only be defined early, or be very prescriptive where this adds value and does not delay a solution.

In DSDM as this definition work is carried out during the Foundations phase of the project, there is the danger that over-specification of quality based on unfounded assumptions may take place. The DSDM guidance does advise that any definition of done work upfront during the Foundations phase should be reviewed regularly throughout the project life cycle, but leadership is required to ensure that this does not encourage BDUF.[viii]

To support Agile Leadership Behavior Two on large projects, there are three important strategies which encourage adaptation to change rather than denial of the need for change. The next three sections explore these strategies:

- ♦ Make decisions at the *latest responsible moment* with *progressive fixity*
- ♦ 'Weave' the requirements and architecture together
- ♦ Focus on the urgent and immediate

Latest Responsible Moment/Progressive Fixity

As an agile project progresses, requirements will become more certain during each iteration of activity. This will reduce the waste associated with continually changing objectives and direction. Progress is impeded by stopping and starting work as targets change. Once a feature is started within an iteration, it should be completed as planned for that iteration. It is tempting to 'tweak' and fine-tune the short-term plan. Encourage discussions about requirements at the start of each iteration of development. Get the team members to agree some discipline in changing the features listed in the sprint backlog to protect them from this problem while the iteration is underway.[ix]

Early decisions on requirements and solution approaches may prematurely rule out better options. As work progresses, the project team becomes more sure about the solution, and committing time to developing the detail becomes less risky. Decisions should be made at the *latest responsible moment* possible. As each decision is made the design becomes more certain. This is the concept of *progressive fixity*.[x] For example, many details of the design of Terminal 5 at Heathrow airport were left as late as possible, until there was greater clarity about the requirements of the new Airbus A380, which was still under development.[xi]

'Weave' the Requirements and Architecture Together

In line with the concept of EDUF, a light architectural framework should be developed which will help large project teams develop solutions with common standards.

However, as elsewhere in agile, it is important not to over specify by deciding on the detail of solutions earlier than the development teams need. Bashar Nuseibeh recommends spending time early on in the project understanding not just stakeholders' requirements but also the technical framework. The idea should be to 'weave' these together to discover requirements and technical constraints/opportunities at an overview level. If the project starts with requirements alone, this invariably results in a waterfall development process. Requirements become artificially frozen in time, and constrained architectures handicap developers by resisting inevitable and desirable changes in requirements.[xii]

Attention to requirements and technical constraints that generate the greatest risks if left unexplored will help address the three major inhibitors to requirements agreement identified by Barry Boehm: [xiii]

- ♦ 'IKIWISI' – 'I'll Know It When I See It' occurs when requirements emerge only after users have had an opportunity to view and provide feedback on models or prototypes.
- ♦ Expecting a packaged solution to be configurable so as to meet a theoretical list of detailed requirements. Experience that Commercial Off The Shelf (COTS) solutions are seldom as easy to use 'off the shelf' as expected. Customers must flex their requirements and make their processing consistent so as to ease COTS implementation.
- ♦ Rapid Change – a feature of the modern world, and if solution development cannot accelerate to match the new technology that is becoming available, then changing technology may wash away development work before it is implemented.

Focus on the Urgent and Immediate

The third major strategy of Agile Leadership Behavior Two is to focus on the solution detail necessary only for the increment that is underway, not on future work. This gives a delivery focus on the immediate timebox of work that will deliver the earliest benefit to the stakeholders. This allows what Rachel Cooper, of Salford University, calls *hard* or *soft stage gates* at the end of each phase. *Stop/go* decisions occur at hard gates, whereas soft gates allow teams to work in parallel, and ensure that the whole project is not stopped dead in its tracks just because of a problem in a single area.

Rules, Regulations, and Managing Change

US government strategy has a specific intent with regard to not just allowing requirements to change, but anticipating and embracing necessary change. In 2011, Teri Takai, the new Defense Department CIO, said one of the biggest challenges is the federal budgeting process and its mismatch with the technology life cycle:

> "We want to look at agile development and deliverables, but the appropriation process wants to know, 'womb to tomb', how much this is absolutely going to cost. That's a very difficult thing for us to say. It starts from a premise that IT projects are engineering projects, without the people dimensions of how you actually get them in and get them to work." [xiv]

Takei identified the following inhibitors:

- Large integrators not having agile developers
- Heavy and slow acquisition practices
- Implementation of agile being treated as a training issue, rather than a culture change issue
- Timing of the change to agile – government technologists, business people and acquisition specialists and the industry need to change together.

The 25-Point Plan recognizes that the funding model whereby Congress approves specific long-term budgets relating to specific outputs is very wasteful. It proposed the set-up of Working Capital Funds (WCFs). These allow each agency the flexibility to re-assign funding for managing flexible IT funding with revolving funds at agency level. Previously, each budget line was ring-fenced in an inflexible manner against tightly controlled budgets.[xv]

Jared Serbu reports that funding has become a thorny issue with some agency CIOs who have identified challenges involved in the OMB reforms. Despite "positive discussions" with members of Congress on the issue, no proposals for legislation had yet been produced on this aspect of the announcement. The key problem, he points out, is that the technology life cycle and the acquisition process are out of phase with each other. He cited various approaches that have been attempted to get around this problem:

- US Customs and Border Protection was moving towards performance-based contracts based on business outcomes rather than detailed specifications of outputs. The aim was to build services, not systems.
- The Department of Veterans Affairs used their pooled budget to fund their 'Transformation 21 Total Technology' (T4) program to provide agile

development capabilities within 14 days rather than 180 days (see the case study in Part I for more about the success of their agile Education Benefits Project).

In the UK the Treasury Green Book guidance has been more problematical with regard to the flexible response to requirements that Agile Leadership Behavior Two is intended to inspire:

- The Green Book requires "clear specification of quality standards" at the earliest stage of the project life cycle and a detailed business case.[xvi]
- It identifies that *procurement risk* exists if the capabilities of the contractor are inadequate, or where necessary changes cannot be easily made. It requires that plans and contracts should assume that change will take place, and should facilitate and harness the advantages of changes. [xvii]
- It recommends the use of complex modeling. This, unfortunately, as we have seen from the PAC's comments above, does encourage spurious levels of apparent accuracy. The focus is on how costs will change according to various variables, rather than on how costs can be capped by cutting back on other variables. An agile approach should be taken when interpreting the advice by illustrating how expected benefits would flex if various features of a project are descoped if costs start to escalate. The focus should be on capping funding and timescales and flexing features. [xviii]
- It assumes that an upward drift of costs will occur as more requirements are identified during a project, and that the only remedy is to do more detailed, up-front requirements analysis. The agile approach does the opposite. Rather than trying to clamp down on *scope creep*, agile welcomes new ideas, as long as less important requirements are traded out. The Green Book advice can tend to encourage the identification of minor features which constrain the technical design.[xix]
- It advises that research should take place as a separate step prior to work starting. It suggests comprehensive research covering theoretical work, such as projected trends and published forecasts, and expected technological developments. It places emphasis on making a large commitment based on a substantial first phase of theoretical research. An agile approach would be to test theory at the earliest possible stage by piloting and by releasing small increments of capability, thus providing practical feedback and breaking decisions on the detail of requirements and on the intended design until the latest responsible moment. If possible, models and demonstration facilities should be used to gain real-life feedback. What has to be avoided is

agreement of a monolithic design at one gated point based on unsubstantiated economic theory. One should operate with progressive fixity – agreeing designs in a modular fashion no sooner than necessary within the framework of a flexible business case that keeps alternative courses of action open so that the approach can flex according to experience. [xx]

- It only requires that key assumptions, options and expected implementation issues should be documented, but does not require these assumptions and options to be kept alive and reviewed during the project. A "Business Case Assumptions Register" should be maintained throughout the life of the project, and periodically reviewed. Where an assumption is found to be incorrect, then the business case should be revisited to decide whether a previously rejected option should be reconsidered. [xxi]

Conclusions

Agile projects seek out change by engaging with stakeholders, carrying out proactive research, and by setting up teams quickly when they are needed based on the people and resources available. They are often *skunkworks* teams that do not stand on ceremony, or worry excessively about their office environment. They just concentrate on doing the work as effectively as possible.

Agile teams do not resist change to project objectives – these are inevitable, and may emanate from policy level – in other words political decision-makers. When administrations change, the shock to the system of what is often a completely different direction must be absorbed. Agile Project Management is an effective way of being able to alter direction when an incoming administration changes policy.

Decisions that will commit to a certain course of action should be taken at the latest responsible moment. Only the immediate problems facing the team that can be solved now are dealt with. Future problems are put onto a backlog of work and dealt with when necessary. This is not procrastination. The solutions to problems are easier to find once related and more immediate problems have been solved. Sometimes problems disappear, and the effort spent in worrying about these at too early a stage is wasted.

Guidance materials, such as the UK Treasury Green Book, have encouraged over-detailed business cases that inhibit the ability of projects to adapt and change. Spurious exactness is exhibited in these business cases, with calculations providing seeming accuracy to several decimal places while assumptions are being used that could be off by a large factor.

Waterfall projects try to suppress change by creating *baselines* of requirements and designs that interlock and depend on each other. The risks of committing to a Big

Design Up-Front (BDUF) approach can be reduced by using research and development activities using mock-ups, working prototypes and the delivery of partial solutions which can be incrementally enhanced over time.

I have discussed how, in the US, one of the biggest challenges is the federal budgeting process which, effectively, encourages BDUF and detailed plans that have to be followed in detail to release funding.

In the UK, the Treasury guidance also causes a problem. Comprehensive product quality plans have to be in place at the earliest stage of the project. The guidance assumes that change is a risk that should be inhibited rather than accepting change as a fact of life that can be harnessed to our advantage. It recommends complex quantitative modeling, which can encourage reliance on assumptions that are subject to great variances, and these models expect budgets to change rather than an agile approach where delivery scope is flexed to fit within budgets.

Change is inevitable in any project. Often a premature commitment is made to a specific course of action in a business case, with no mechanisms to regularly review the assumptions on which the decision was made. Resistance to change is the norm. The current strategic best practice in both the US and in the UK emphasizes the need for upfront, detailed outcome modeling and sensitivity analysis in business cases. This creates a natural resistance to considering alternative courses of action later on.

Agile leadership is about seeking out necessary changes to existing plans so as to reduce risk, and then harnessing these changes to enhance returns. When necessary changes have a major impact on the business case, it is better to embrace them early in development.

Agile approaches help harness change for advantage. Risks can be seen as opportunities to be proactive and engage with potentially disruptive stakeholders at an early stage. Be vigilant against activities that tend to create inertia and project inflexibility. Chief among these are over-large specifications and designs which have been agreed in too much detail before enough experience has been gained through practical development activities.

Externally focused activities, such as specification prototyping and modeling, and lightly governed skunkworks teams will help reduce the risks of introverted behavior that simply ignores risks and reduces the possibility of identification of unexpected opportunities.

[i] {Great Britain. Treasury 2003 #182}
[ii] {UK HM Treasury 1991 #183} and {Great Britain. Her Majesty's Stationery Office 1997 #181}
[iii] {What price a PFI project 16 June 2002 #185}
[iv] {NAO 2003 #186}
[v] {Beath 1994 #70: 370}. The materials they reviewed were from the writings of James Martin, discussed further in Part III.
[vi] {Gomaa 1981 #190}
[vii] {Skunk Works Lessons Learned 1997 #176}
[viii] {Craddock 2012 #330}
[ix] {Williams 2012 #239}
[x] {Cooper 1994 #177}
[xi] {Davies 2009 #109}
[xii] {Nuseibeh 2001 #179}
[xiii] {Barry Boehm 2000 #180}
[xiv] {Serbu 2011 #346}
[xv] {Kundra 2010 #157: 23–24}
[xvi] {Great Britain. Treasury 2003 #182: Box 23}
[xvii] {Great Britain. Treasury 2003 #182: Box 4.3}
[xviii] {Great Britain. Treasury 2003 #182: Section 5.17}
[xix] {Great Britain. Treasury 2003 #182: Box 4.2}
[xx] {Great Britain. Treasury 2003 #182: Para 3.4}
[xxi] {Great Britain. Treasury 2003 #182: Para 6.15}