

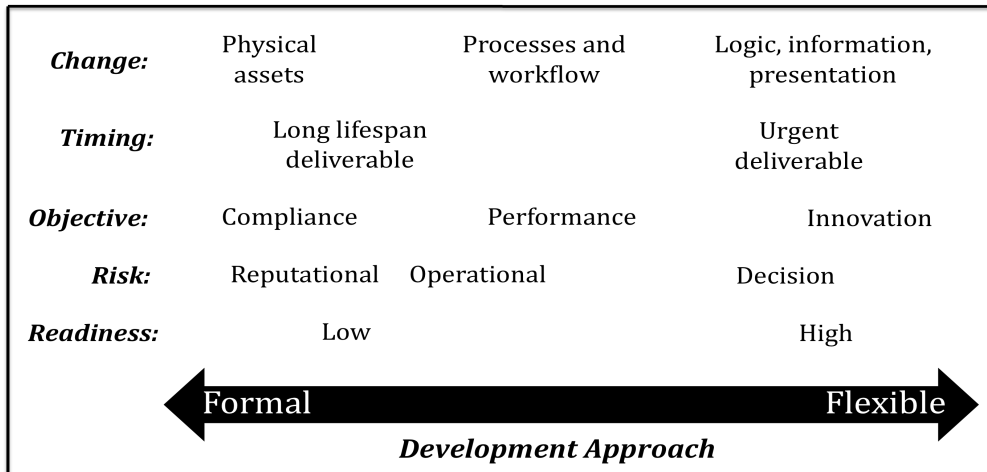
## Drivers of Applications Development Approaches

### Bob Morison – April 2013

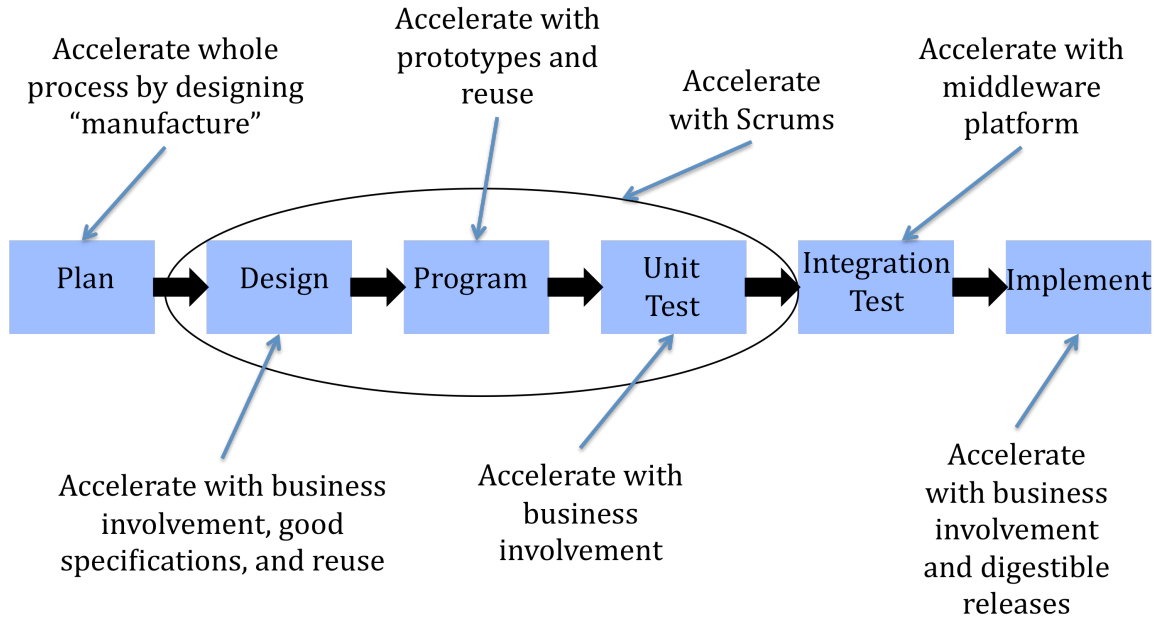
Last week's point of view contrasted the operating models for "formal" (think Waterfall) and "flexible" (think Agile) applications development. This week, I'd like to discuss what we call (in the "Third Way Applications Development" research) the *drivers* that move a project toward more formal or flexible development and the *accelerators* that can move any project along. Five basic drivers are summarized in the diagram. Each centers on a question:

- **What's being changed?** If the change centers on information and its use, flexible approaches generally work well. If business processes and people's workflows are changing, more formality is needed. And if changes to expensive physical assets (such as business equipment and facilities) are involved, then development – especially implementation – needs to be formal still.
- **What's the timing?** Urgent deliverables naturally call for fast and flexible methods, though it may be necessary to adjust the scope of the deliverable, or divide it across multiple releases, to get the necessary speed. Applications expected to run reliably over long lifespans demand more formal development.
- **What's the dominant objective?** If regulatory or another form of compliance is paramount, formal approaches are called for. If you're after innovation, you need the flexibility to experiment. If the goal is improving business performance, you're somewhere in the middle.
- **What are the risks?** If the business changes represented by the new application involve primarily decision risk (where mistakes may be made but you can recover quickly), flexible development is appropriate. If the risk is more operational – or especially reputational – formal approaches are needed.
- **What's your readiness?** If the type of project is familiar, and you have the experience and skills to deliver confidently, then the approach can be flexible. Under the opposite circumstances, it's better to take a more formal approach – but capture your experience so you can be more flexible on the next similar project.

The drivers behind any given initiative may well pull you in different directions. For example, mobile applications for customers benefit from experimentation and iteration to optimize the customer experience – calling for flexible methods and field testing. However, any major problems with the applications, especially misuse of customer information, can incur both operational and reputational risk – calling for rigorous testing and review before large-scale rollout.



Once you've adjusted a project's development approach based on the drivers, adjust it again with accelerators in mind: What's helped with similar projects in the past? What specific things have we done that really improved both speed and performance? What steps can we speed up? How soon can we prototype? The next diagram lists some of the most basic accelerators. Incorporate your own, and make them as specific as possible.



When projects are under way, and especially soon after completion, take time to capture your experience. Then use it to refine your understanding of the drivers and accelerators next time around.