

Alexander Babkin

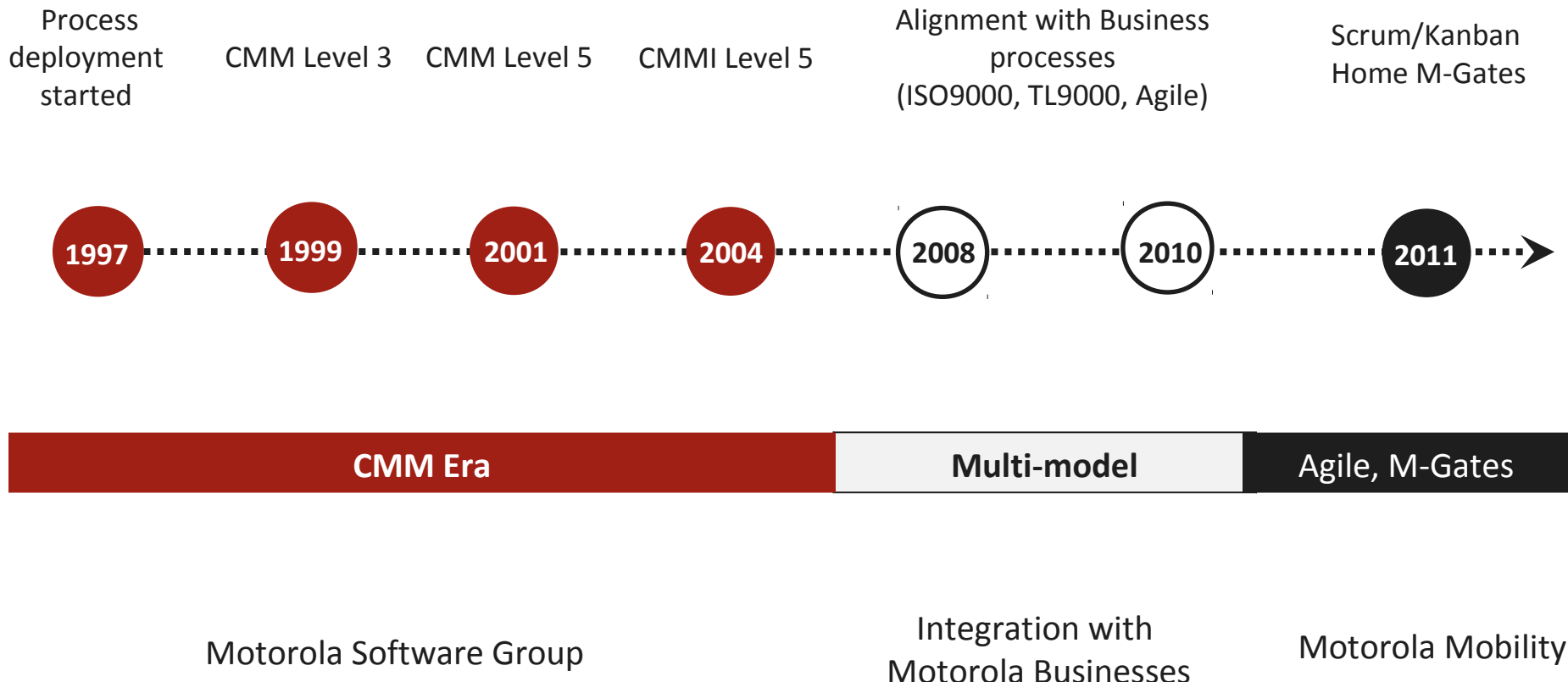
Quality Engineering Manager
Motorola Mobility

<http://ru.linkedin.com/in/alexanderbabkin>

Applying Agile in the Corporate Environment



Process Background



Corporate Environment Factors

- Legacy process
- Org structures
- Distributed teams
- Distributed expertise

Why go AGILE?

Competitive **advantage** through *faster* results
driven by immediate *customer's feedback*



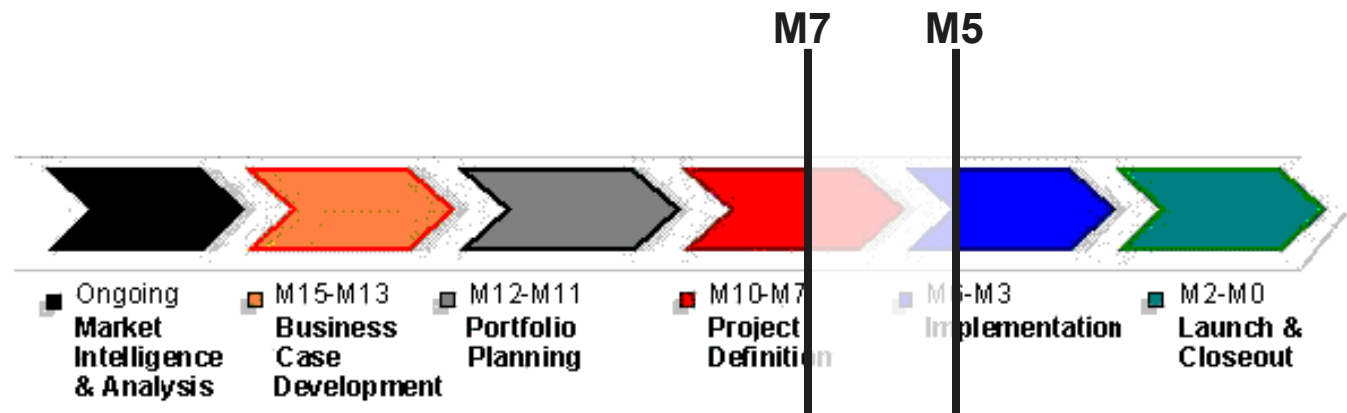
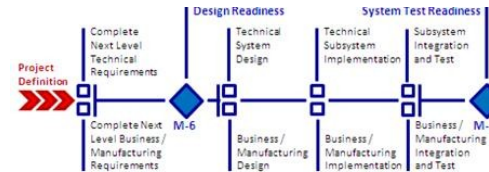
Agile under corporate factors pressure

- ***Scrum*** vs. Corporate Processes Framework
- ***Product Owner*** vs. Customer vs. Product Manager
- ***Scrum Master*** vs. Project Manager
- ***Self-organizing Team*** vs. Org Structure
- ***Cross-functional Team*** vs. Roles



Corporate Process Framework (M-Gates)

- High-level description of product lifecycle
 - 16 Gates and 5 Phases
 - From idea of a product till retirement
 - Tasks of various functional teams,
 - Collaboration & communications



- Use of Agile constrained by M-Gates pass criteria
 - e.g. when a product passes to system or field testing it becomes harder to introduce changes and cycle of customer's feedback becomes longer.



Product Owner vs. Customer or Product Manager

- In the corporate environment
 - Not an end customer but rather technical manager
 - Plays dual role of managing product requirements and overseeing software development
- New role and especially important for distributed development within corporate process framework.
 - Fills in a gap between End users/Marketing and development
 - At the same time should be a single point of contact for big product/system



Product Owner vs. Customer or Product Manager

- The following Risks are observed:
 - Case 1: Product Owner stay aside of Scrum team
 - Case 2: No single authorized Product Owner
 - Case 3: Micromanagement
 - Case 4: Cases 2&3 combined
 - Case 5: Agile flexibility misinterpretation



Product Owner vs. Customer or Product Manager

- Mitigating Risks:
 - Kick-off trainings for Product Owner to ensure he/she shares agile values in the same way as Scrum teams.
 - Assignment of Lead Product Owner role
 - maintain overall product vision and product backlog
 - manage dependencies among distributed Scrum teams
 - Scrum Master may take over some responsibilities of Product Owner
 - typically, identification and tracking dependencies and with other Scrum teams
 - Retrospectives for Scrum of Scrums
 - Retrospectives with Product Owner



Scrum Master vs. Project Manager

In general, we **didn't succeed** in **distinguishing** these roles

Self-organizing Team vs. Org Structure

Yet teams are **not** truly **self-organized**



Why “PM” and “Org Structure” prevailed

- In Corporate, an employees is typically **developed** to become a manager:
 - Regular Capability & Skills Assessments, Managerial Ladder
 - Similar professional characteristics: decision making, communication, big picture view etc.
 - Why this candidate Scrum Master hasn't become a Manager yet?
- Need for timely conflict resolution
 - Motivation of team members, e.g. “unacceptance of agile”
 - Task assignment due to different qualification/expertise
 - Debates on technical solution
- Need for protecting a team from external interferences
- Corporate Performance Management System
 - Focused on individual performance of an employee (not team performance)
- Legacy Reporting Processes



Other side of management-driven approach

- **Observed Risks**

- Teams may tend to over-commit under management pressure
- Less team motivation => less accountability for results
- Less fun!



Toward Self-Organization...

- **Mitigating Risks:**

- PMs are trained and encouraged to apply soft styles of management
 - “I would recommend you to do it this way...”
- Training and coaching of knowledge sharing and team self-consciousness
- E.g. “Rules of engagement”
 - *Review design idea with your colleague*
 - *Review code with your colleague*
 - *When you’re ready help someone else before taking a new story*
 - *Use pair-programming as appropriate*
 - *Weekly sharing*
 - *Leave code cleaner than you found it*
- Most of PMs of Agile projects continue to do some coding...



Cross-functional teams vs. Roles

- **Case 1:**

- Agile deployed over legacy processes
- Historical distribution of expertise around the globe
 - development vs. testing
 - architecture, UI, middleware, testing
- Cost to pay:
 - (N+1) sprint for testers
 - more formalities (documentations) to enable knowledge sharing
 - Scrum of Scrums to specifically address test activities coordination

- **Case 2:**

- New project driven by newly acquired team (no corporate background)
 - expertise is equally distributed across teams
 - knowledge sharing thru daily Scrums
 - collocation of teams
 - collocation was named as one of key factor for Agile success by one of the Program Managers



Transformation of other Agile (Scrum) practices

- Sprints are typically 2-3 week long to enable faster delivery & change incorporation
 - Examples of Kanban use to deal with continuous changes
- *N+1* test sprint for projects with *non-located* test teams
 - With additional 1-2 day for test sprint planning
- Lead Product Owner for dependencies of requirements allocated to distributed teams
- Sprint Planning exceeds 1 day in projects with complex dependencies with other teams
- Daily Scrum *rarely* may involve discussions of dependencies with other teams
 - Trying to avoid this as much as possible and handle them in off-line



Conclusion

- **Agile** practice **proved** their **value** despite all challenges we faced deploying them in corporate environment
 - Faster results, faster reaction to changes
 - Products developed with use of Agile received positive feedback from consumers
- Still we have challenges to work with and keep on experimenting to balance Agile with 'legacy' processes:
 - E.g. Configuration management, test management tools, marketing requirements management, design & user documentation, metrics, program reviews, customer surveys
 - Agile trainings for team members
 - Seminars to share about processes, tools and success stories of Agile use
 - Technical seminars for engineers to share technical expertise
 - Global workshop to share & learn from external teams (e.g. India, China)
 - Management town-halls to inspire engineers



“Corporate” Agile Manifesto? 😊

- **Individuals and interactions over processes and tools**
 - *and we have mandatory processes and tools to control how those individuals (we prefer the term ‘resources’) interact*
- **Working software over comprehensive documentation**
 - *as long as that software is comprehensively documented*
- **Customer collaboration over contract negotiation**
 - *within the boundaries of strict contracts, of course, and subject to rigorous change control*
- **Responding to change over following a plan**
 - *provided a detailed plan is in place to respond to the change, and it is followed precisely*

That is, while the items on the left sound nice in theory, we’re an **enterprise company**, and there’s **no way we’re letting go of the items on the right.**

<http://www.halfarsedagilemanifesto.org/>



Thank you! Questions?

