InterDigital®

May 9, 2012

Architecture as the Glue:

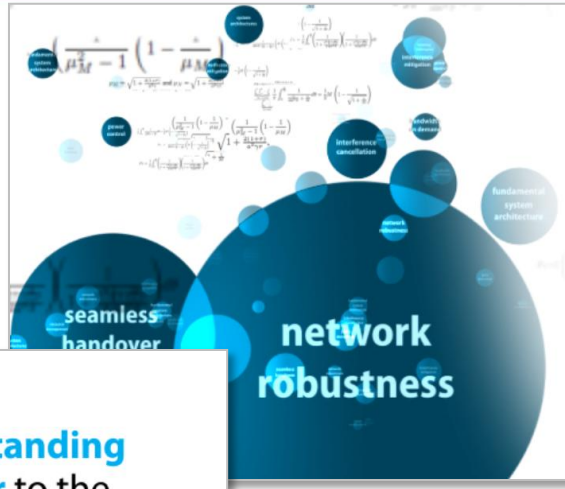
Lessons Learned Organizing Multiple Agile Teams

Robert Gazda, InterDigital, LLC
Steve Thiffault, InterDigital, LLC

- **What are we covering?**
  - Our experience is using a ***"Shared" and "Understood" Architecture*** as an essential tool, needed to facilitate the organization and management of ***Multiple Agile Teams*** (multi-location / time-zone, multi-cultural, etc) in building a **Single Software Product**.

- **Outline**:
  - **Context:** InterDigital + Smart Access Manager Introduction.
  - **Methods Description:** Our use of Architecture *"As the Glue"* to organize Multiple Agile Teams.
  - **Lessons Learned:** What worked, What didn't.

About InterDigital



InterDigital develops **fundamental** wireless technologies that are at the **core** of **mobile devices**, **networks**, and **services worldwide**.

HSPA+
LTE
LTE-A
802.16
802.21

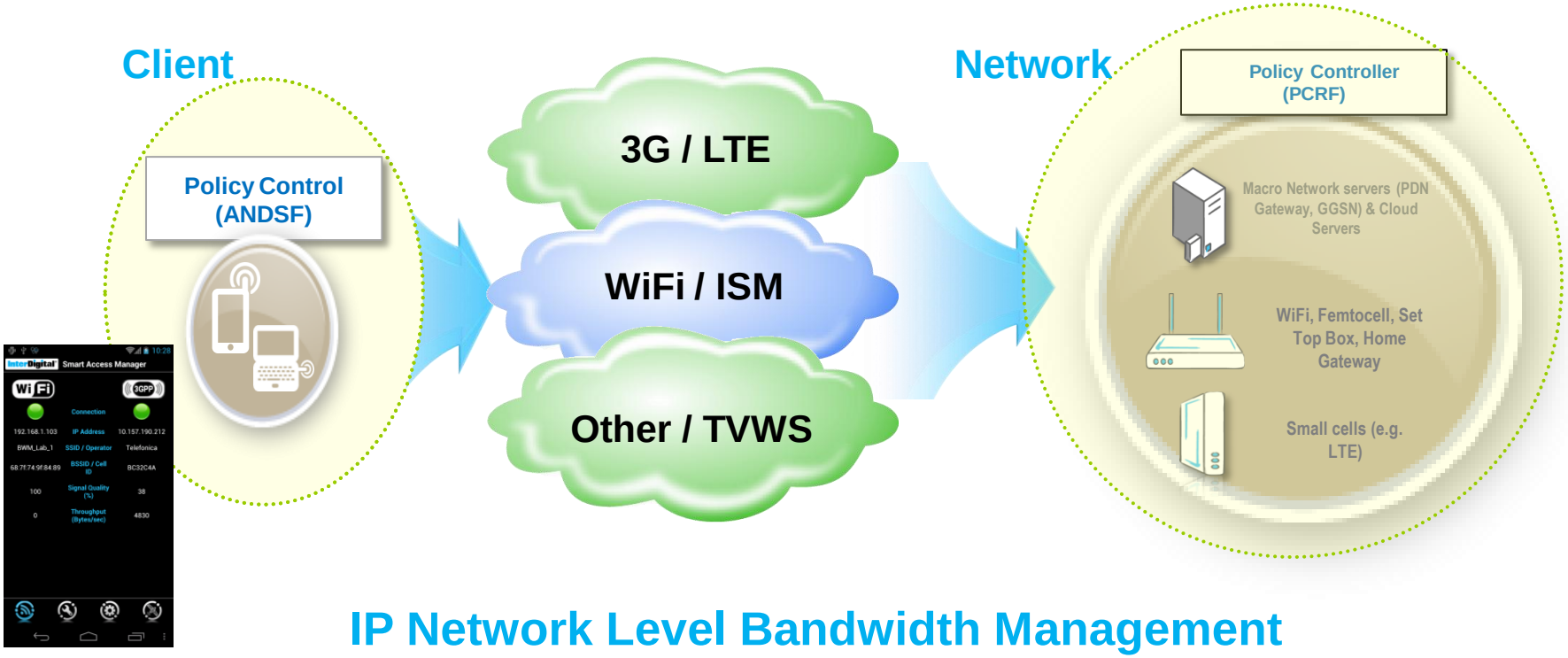
As a **long-standing contributor** to the wireless industry, we solve many of the most **critical** and **complex technical challenges years ahead** of market deployment.

Our advanced solutions support **more efficient wireless networks**, a **richer multimedia experience**, and **new mobile broadband capabilities**.

InterDigital®

Intelligent Connectivity Middleware for Mobile Devices

Part of a Holistic Bandwidth Management Solution Suite



IP Network Level Bandwidth Management

- Adaptable Policy driven network and application flow management
- Supports IP Flow Segregation, Aggregation, and Mobility
- Context, Application, Service Level — Aware
- Flexibly configured, integrated Client – Server Solution

Product Scope Growth + Team Growth

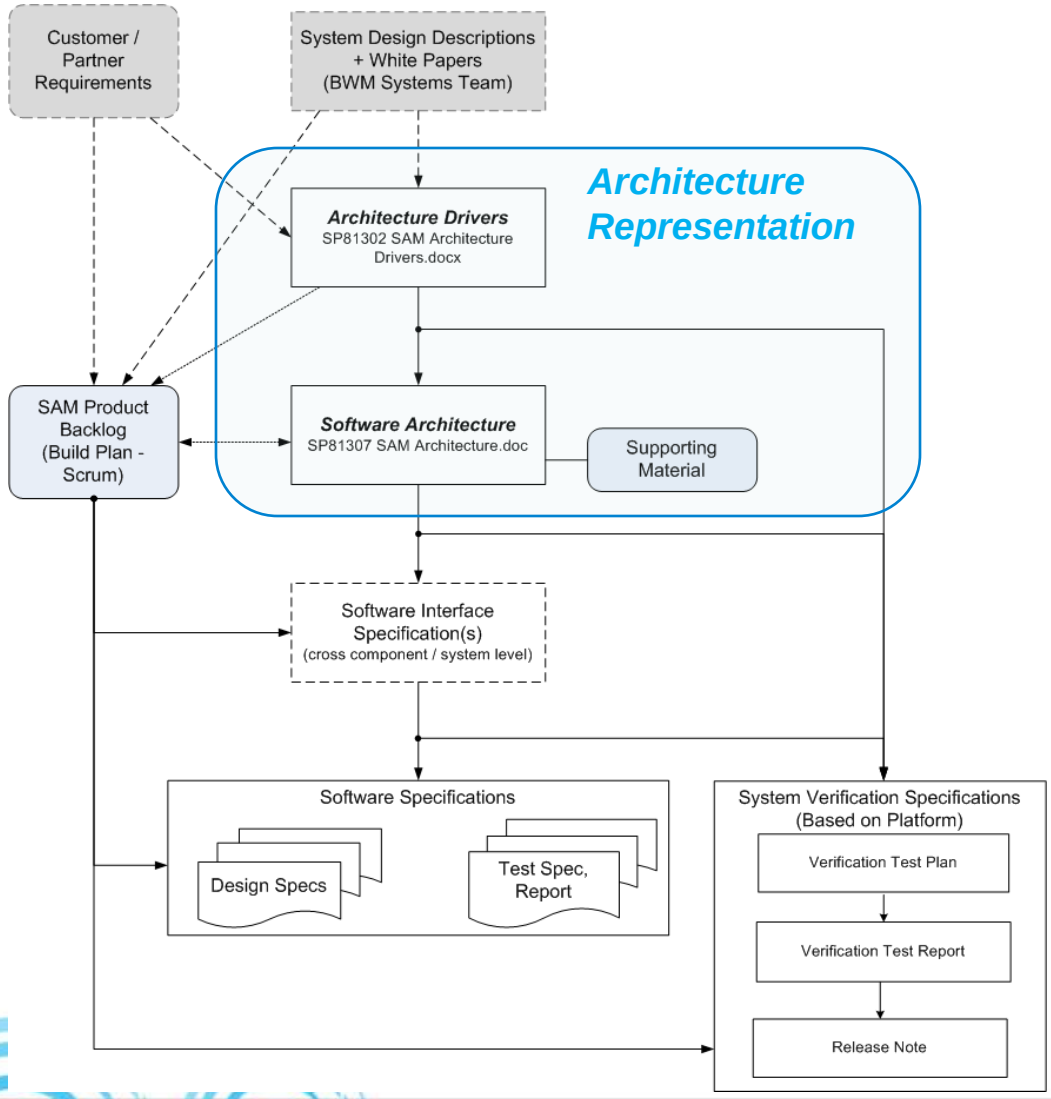
- **Where did we start?**
  - Research and Innovation Project → Lab Only Demonstrator.
  - Functionality was all that mattered.
  - Small Team: 4-5 Engineers. Organized as a Single Scrum Team.

- **Where are we now?**
  - Interoperability tested with several Network Vendors.
  - Beginning Operator field testing and trials.
  - Expect end-customer deployments to start towards year
  - Functionality and Quality Attributes count.
  - Larger / Distributed Development Team:
    - ~30 Engineers
    - Three locations (Montreal, King of Prussia, Gurgaon)
    - Organized into 5 Scrum Teams – "Scrum of Scrums"

InterDigital®

Architecture Representation

Document Tree



Architecture Drivers:

- Requirement sub-set, effecting the architecture structure.
- Includes: Functional Requirements + Constraints + Quality Attributes
- Drives Product Backlog (Scrum)
- Created from Customer / Partner and Systems Engineering Input

Software Architecture:

- Based on 4+1 Architecture Views.
- Key Views: Context Definition (External Interfaces), Logical View, Physical / Deployment View, Run-time View, and Scenarios
- Architecture drives the Backlog…Backlog drives the Architecture

Pre-Scrum or Release Start-up ("Sprint Zero"):

- Architecture Drivers and Initial / Concept Architecture
  - Created from Customer / Partner / System Engineering input – small team of Senior Developers
  - Drivers and Architecture refined against each other.
- ***Architecture and Drivers →***
  - ***Use to create the "Initial" Product Backlog for Scrum.***
- Scrum Kick-off ("Sprint One") →
  - Drivers, Architecture and Backlog analyzed, reviewed, and agreed with all stakeholders.
  - Light-weight architecture analysis performed as part of the analysis.

During Scrum / Sprint-to-Sprint:

- Architecture is the ***"Shared Vision"***, used to organize development and integrations.
  - Blue-Print to organize and "charter" Scrum Teams.
  - Critical when features span multiple Scrum Teams.
- But…Architecture is ***Not*** **Static, under constant Refinement**:
  - Product Backlog includes "Analysis or Feasibility" Backlog Items.
  - Executed by Senior Developers when assigned to drive new feature development.
  - Output == Architecture Refinement + "Production" or development Backlog Items.
- Architecture is ***Shared*** across all Scrum teams → ***No "Chief" Architect***.
  - Senior Developers in each team tasked to evangelize architecture within each Scrum team.

Lessons Learned – The Good…

1. Architecture as a "Blue-Print" – Conway's Law

- Designed the Organization Structure around the Architecture.
- Maintaining alignment both in the Team and Architecture.
- Reaffirmed Conway's Law.

2. Shared Architecture – Everyone Owns it.

- Project staffed with a core group of Senior Engineers from the start.
- All teams may change the Architecture (driven by Senior Engineers).
- But, all changes must be communicated, reviewed, and agreed by all teams.
- Established Trust.

3. Loose Coupling is Good: Features / Interfaces / Sub-systems

- Allows both the software modules and the Scrum teams to work as independently as possible.
- Organize development around features, when practical.

4. Grow the Architecture as the Product Grows

- Architecture is “Alive and Kicking”
- Refine architecture as the requirements, features and constraints change.
- Better than “Big Up Front Design”
- Architecture and Product Source Code under constant analysis and refinement - “Sprint to Sprint”.
- Trust that the team can react to change.

Lessons Learned – The not so Good (i.e. the Bad)…

1. Cross-Site + Multiple Team Development is Hard

- Difficult to align the development of multiple teams
  - Especially, cross-site / time-zone.
- No silver bullet to make cross-site / multi-team development easy.
- Architecture is better understood and communicated internally than offshore.
- *Attempting to bridge the offshore gap with cross-located Scrum team members: one in India and one in Montreal.*
- *Stress a consistent and understood definition of Backlog Item "Doneness" in all teams.*
- *Improve "Scrum of Scrum" effectiveness - inter-team awareness.*

Lessons Learned – The not so Good (i.e. the Bad)…

2. Architecture Drivers static compared to the Architecture.

- Driver Specification is used at the start and than falls to the side, becoming stale.
- Drivers ended up "living" in the Product Backlog.
- *Update Drivers at major milestones (e.g. Scrum Release time boxes).*

3. Using Quality Attributes to Verify the Architecture

- QA analysis performed in a limited manner at project start (lab only scenarios).
- End-customer QAs's are now growing in importance.
- *Drive QA Scenarios in the Product Backlog and capture them in the Architecture Drivers Specification.*
- *Refactor the SAM Architecture and Implementation, "as needed".*

Thank you!!

InterDigital®