



IoT meets Agile

Rev. 2.0.0 – Settembre 2024



AgileIoT

www.agileconstellation.info

AgileIoT

an AgileConstellation Star



Sommario

Introduzione.....	6
1. Il mondo dell'Internet of Things.....	7
1.1 Cos'è l'Internet of Things	7
1.2 I principali campi di applicazione	8
1.3 Le sfide: privacy, security ed energy saving	10
2. La governance dei Progetti IoT	11
2.1 L'ambito multidisciplinare	11
2.2 L'aspetto hardware	11
2.3 L'aspetto software	12
2.4 L'aspetto cloud	13
2.5 La complessità della governance	13
3. AgileIoT.....	15
3.1 AgileIoT: an AgileConstellation Star	15
3.2 AgileIoT Fast Prototyping	17
3.3 I Framework	18
4. AgileIoT Duttile	19
4.1 Prototyping Phase.....	20
4.2 Engineering Phase.....	22
4.3 Workout Phase	24
4.4 Working Team.....	25
4.4.1 Primary Roles.....	26
T-Shaped Makers	26
Prime Maker	27
Product Owner.....	27
4.4.2 Closer Roles.....	28
Energy Closer	28
Data Flow Closer	28
Cloud Closer.....	28
Security Closer	28
Legal Closer	29



4.4.3	Temporary Team.....	29
	Fast Prototyping Temporary Team.....	29
	Story Temporary Team	29
4.5	Artefatti	30
	Work Item.....	30
	Epic.....	30
	Makers Story	30
	Task	31
4.6	Visual Management.....	32
	Product Backlog.....	32
	Makers Backlog	32
	Makers Board	33
	Technical Debt & Risk Board	33
	Burndown Chart	34
4.7	Delivery Item	34
	Smart Thing Bill Of Materials	34
	Smart Thing	34
	Solution to Delivery.....	34
	Solution Definiton of Done Document.....	35
4.8	Eventi	35
	Vision Fast Prototyping	35
	Planning	36
	Backlog Planning	36
	Refinement.....	36
	Iteration Flashback	37
	Retrospective	37
	Solution Overview.....	37
	Technical Retrospective	38
	Methodology Retrospective.....	38
4.9	Metriche	39
	Rhythm.....	39
	Lead Time & Throughput	39
5.	AgileIoT Fiotto	40



5.1	Work Item	40
	Epic	41
	Makers Story	41
	Task.....	42
5.2	Fiotto Team	42
5.3	Aree	43
5.4	WorkPivot	43
5.5	D-ARCH.....	44
5.6	D-ARCH Metriche	45
6	Conclusioni.....	48
	Author.....	49
	Co-Authors.....	49



Introduzione

Benvenuti nel mondo di **AgileIoT** e dei suoi due framework operativi **AgileIoT Duttile** ed **AgileIoT Fiotto**.

AgileIoT guarda in modo *agile* ai progetti del mondo dell'*Internet of Things* (IoT), proponendo una serie di strumenti atti alla loro governance ed esecuzione.

Sia AgileIoT Duttile che AgileIoT Fiotto abbracciano la *Filosofia, i Principi e le Pratiche* dell'**AgileConstellation Manifesto**¹, fornendo al Team due framework per lavorare in chiave *Continuous Value* e *Continuous Deployment*, supportati dalla logica “pull”.

Base di partenza sono i successi raggiunti in ambito Agile e Lean negli ultimi decenni, adottando però un occhio critico che tiene conto di quelli che sono stati i principali problemi, soprattutto nei tentativi di applicare gli stessi concetti al mondo dell’hardware.

In particolare, *AgileIoT Duttile* fornisce una soluzione olistica, suggerendo un approccio in cui i diversi elementi vengono considerati parte di un unico processo integrato. Non si tratta quindi un over-process framework che cerca di cadenzare le attività di team specializzati nello sviluppo di software e hardware, al fine di creare un *planning* per i momenti di *big-bang integration*.

Dualmente, *AgileIoT Fiotto* fornisce uno strumento di *Visual Management* che tiene conto delle specificità dei progetti IoT e consente di avere una visione veloce e completa dell’andamento del progetto e dei problemi afferenti. L’utilizzo di AgileIoT Fiotto consente di ottenere forti benefici nella gestione complessiva del progetto, avendo prontezza degli aspetti complessivi di realizzazione e quelli specifici dei singoli elementi di Valore.

In entrambe le soluzioni il relativo *Working Team* è un team “reale” che lavora a stretto contatto giornalmente e che ha obiettivi comuni e condivisi. L’obiettivo è quello di sviluppare soluzioni end-to-end, focalizzandosi sul *Valore* da creare per gli stakeholder.

¹ www.agileconstellation.info



AgileIoT

www.agileconstellation.info

1. Il mondo dell'Internet of Things

1.1 Cos'è l'Internet of Things

Con **Internet of Things (IoT)** si intende la possibilità di connettere tra di loro oggetti e luoghi concreti, sfruttando le infrastrutture di rete esistenti. Ciò li rende capaci di comunicare e adattare il proprio comportamento in funzione degli eventi e dell'evoluzione di contesto.

Nell'universo IoT troviamo fondamentalmente due elementi portanti: gli **Smart Thing** ("T", gli oggetti intelligenti) che sono connessi e gestiti dal **Cloud** ("I", Internet) il quale si occupa, inoltre, dell'elaborazione e dell'analisi delle informazioni. Con Cloud si intende l'approccio meta-architetturale, che può sposarsi sia con il *public* che con il *private Cloud*.

Per quanto la terminologia sia recente, l'idea di device connessi ad una rete per avere un flusso costante di dati può essere fatta risalire fino al 1982, quando la **Carnegie Mellon University** modificò un distributore di Coca-Cola per connetterlo in rete al fine di conoscerne, in tempo reale, la scorta di bevande. La prima formalizzazione di Internet of Things risale invece al 1991, quando *Mark Weiser* scrive l'articolo "The Computer for the 21st Century" nel quale immagina un futuro in cui *"hardware e software, connessi da cavi, onde radio e infrarossi, saranno così ubiqui che nessuno ne noterà la presenza"*.

Si tratta, quindi, di un'idea che ha oltre 30anni, ma che solo oggi, con l'evoluzione tecnologica, la banda larga, il Cloud e la penetrazione dell'informatica in ogni ambito della nostra vita, si sta realmente concretizzando. *Gartner* stima che entro il 2020 nel mondo esisteranno 26 miliardi di dispositivi connessi in Rete, mentre il *Pew Research Center* ha evidenziato (survey 2014) che l'83% degli esperti concorda che entro il 2025 il modo in cui percepiamo la Rete verrà completamente subordinato all'IoT, con l'esplosione dei principali settori di applicazione: *domotica, robotica, wearables e industria automobilistica*.



AgileIoT

www.agileconstellation.info

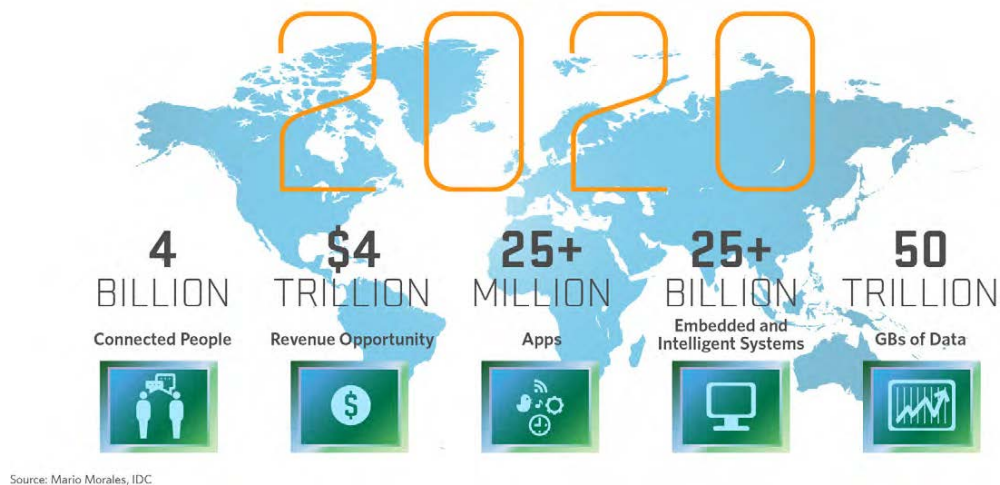


Figura 1 - IoT nel 2020, secondo un'ulteriore studio di IDC

1.2 I principali campi di applicazione

Smart Car, *Smart Home* e *Smart City* sono i principali ambiti di crescita per l'IoT, il che evidenzia la vocazione multidisciplinare in funzione dell'obiettivo che si intende raggiungere.

Si pensi al caso della Corea del Sud: a *Songdo*, 65Km dalla capitale Seoul, è nata la prima *Smart City* del mondo, una città interamente connessa che raccoglie continuamente dati afferenti a decine di ambiti, elaborandoli costantemente senza la necessità di intervento umano. Anche il progetto integrato della città di *Santander*, in Spagna, è particolarmente interessante: circa il 10% della popolazione utilizza un'App che sfruttando sensori distribuiti in tutta la città, fornisce informazioni in tempo reale su *inquinamento*, *parking*, *traffico* ed *eventi*.

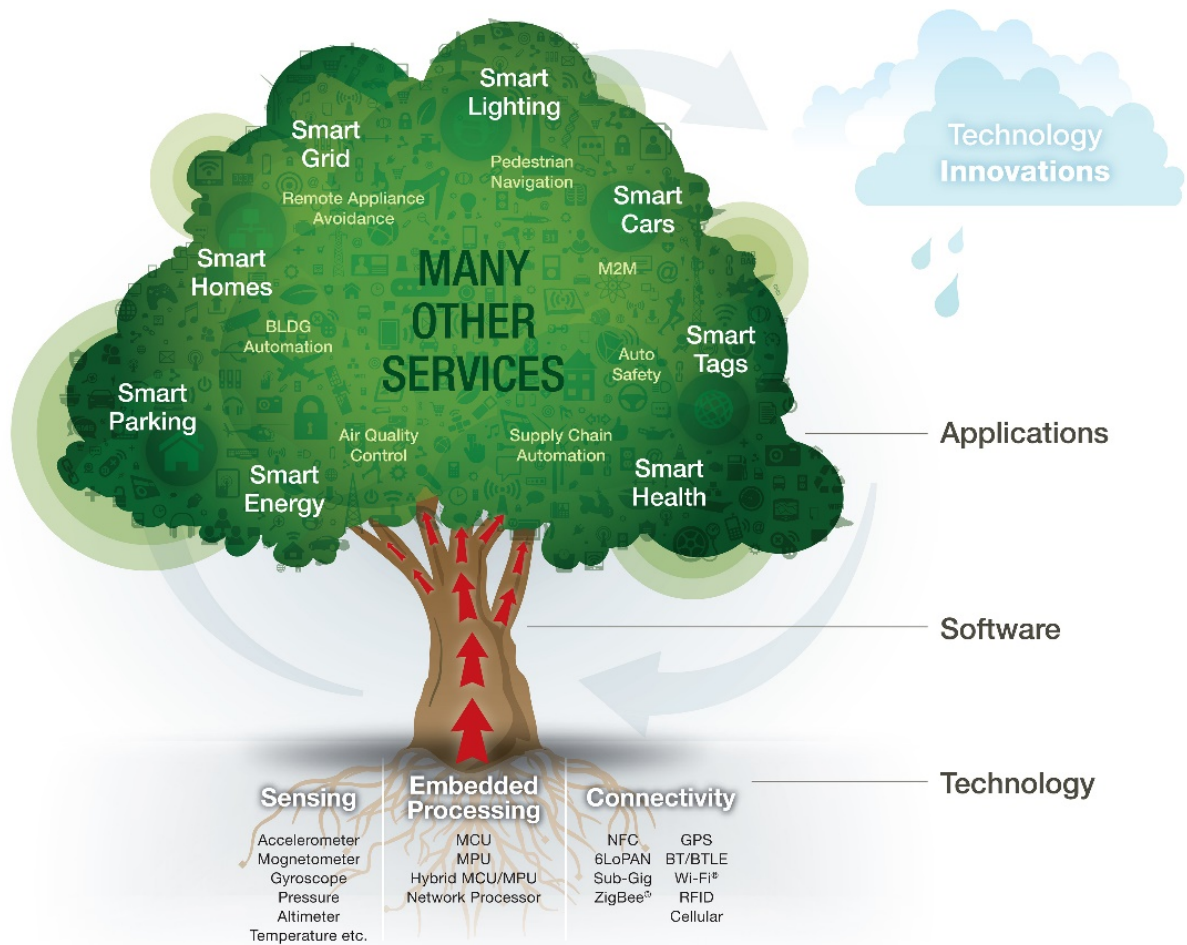


Figura 2 - Campi di Applicazione



AgileIoT

www.agileconstellation.info

1.3 Le sfide: privacy, security ed energy saving

La **privacy** è uno dei grandi problemi legati all'IoT: più device connessi significa più dati personali che viaggiano in Rete. Ciò impone di focalizzarsi sin da subito sulle implicazioni inerenti la sfera personale a cui le soluzioni IoT possono portare.

Strettamente legato a tale aspetto è anche la **security dei dati** di controllo e gestione degli apparati, poiché essi impattano direttamente sul mondo fisico: si pensi, ad esempio, ai disastri a cui può portare un attacco ai servizi di gestione dei computer di bordo delle autovetture.

Se questi primi due elementi sono già evidenti e affrontati da tempo, quello dell'**energy saving** è invece meno scontato, poiché si tende erroneamente a dare per scontato che i dispositivi non abbiano problemi energetici. Cosa accade, però, se il dispositivo di riferimento è, ad esempio, su una piattaforma marina in cui l'energia è limitata e l'ottimizzazione va spinta a livello di pochi wattora? È evidente come l'aspetto energetico possa diventare un driver imprescindibile per la progettazione e lo sviluppo dell'intera soluzione IoT.



2. La governance dei Progetti IoT

2.1 L'ambito multidisciplinare

Una soluzione IoT afferisce a **diversi domini tecnologici**, ognuno con le proprie peculiarità e con le proprie regole, esplicite e implicite. Così, lo specifico progetto deve confrontarsi quotidianamente con gli aspetti legati all'*Hardware*, che può contare su processi maggiormente collaudati e predittivi, al *Software*, dove la complessità e il cambiamento sono all'ordine del giorno, e con il mondo del *Cloud e dei Big Data*, con le proprie regole e i propri elementi di rischio. Oggi, tipicamente, la realizzazione di una soluzione ibrida, hardware e software, avviene in modo parallelo per poi sincronizzarsi in un determinato momento e dare il via alle successive fasi di integrazione e test.

Si tratta di un approccio che sposa la logica della *Big Bang Integration* e che soffre di diversi problemi critici:

- *l'efficienza di sviluppo è fortemente penalizzata, poiché i diversi team non comunicano costantemente tra loro e non condividono il know-how e gli elementi di forza e debolezza;*
- *i problemi vengono alla luce solo in una fase avanzata di realizzazione della soluzione, con costi di fix elevati, soprattutto per la componente hardware;*
- *la soluzione finale spesso contiene un compromesso qualitativo a ribasso, dettato dalla necessità di consegnare la soluzione in tempo e basato su workaround software laddove i vari componenti non si integrino perfettamente.*

Nel prosieguo si andranno a dettagliare gli aspetti primari dei diversi domini.

2.2 L'aspetto hardware

Lo sviluppo della **Componente Hardware** segue tipicamente un approccio waterfall-like, con un processo lineare ottimizzato nel corso degli anni e rigido rispetto ai cambiamenti in corso d'opera. Anche i ruoli e gli skill sono maggiormente definiti, contemplando:

- *il progettista hardware*, che generalmente ha il compito di realizzare lo schema elettrico, stabilendo, ad esempio, che il processore sia collegato alla RAM ed alla FLASH, che ci sia bisogno di un traslatore di livello, di una resistenza qui ... di un condensatore lì, ecc.
- *lo "sbrogliatore"*, che trasforma lo schema elettrico (per il quale i componenti non occupano posizioni "fisiche" su una scheda) in schema fisico corrispondente, nel quale, al contrario, i componenti sono posti sulla scheda reale posizionando i connettori in corrispondenza dei buchi del disegnatore meccanico. Il suo compito comprende,



primariamente, la definizione delle “piste” che collegano i componenti. Dallo “sbroglio” nasce la BOM (Bill of Materials), in quanto sono usate librerie di componenti reali;

- *il disegnatore meccanico*, che definisce la “meccanica” del dispositivo. Ad esempio si occupa del contenitore, delle posizioni dei “buchi” per i connettori alle dimensioni;
- *lo sviluppatore firmware (driver e Board Support Package)*, che si occupa di realizzare i driver di basso livello per le periferiche e, nel caso di utilizzo di un sistema operativo su dispositivo embedded, si occupa di generare l'immagine personalizzata del sistema operativo. Questo vale soprattutto per sistemi basati su Windows Embedded Compact e Linux. Da sottolineare che, tipicamente, lo sviluppatore firmware ha profonde conoscenze hardware poiché svolge anche il ruolo di collaudatore dell'hardware prodotto. Con una serie di test valida il corretto collegamento dei fili ed il lavoro dello sbrogliatore e/o del progettista nella misura in cui essi non abbiano sbagliato qualcosa durante la cablatura (ad esempio, nel posizionare un pin di I/O del processore sul pin di alimentazione della memoria);
- *lo sviluppatore software*, che si occupa dello sviluppo della parte applicativa del firmware, interagendo con i dispositivi (grazie ai driver a disposizione), effettuando elaborazioni, trasmettendo e ricevendo dati, ecc.

2.3 L'aspetto software

Lo sviluppo della **Componente Software** è oggi tipicamente governata dai Valori e dai Principi Agili² utilizzando un mix di pratiche per massimizzare il valore della soluzione prodotta. L'approccio è *iterativo ed incrementale*, al fine di consegnare rapidamente al cliente versioni progressivamente sempre più complete del prodotto ed ottenere velocemente feedback per riallineare le successive attività anche in funzione alla maturazione del team. Un ulteriore aspetto da sottolineare è la vocazione cross-functional dei membri di un Agile team, ovvero developer in grado di intervenire sull'intero progetto senza che parte di esso sia vincolato ad una figura specifica. A supportare gli sviluppatori possono poi essere presenti ulteriori figure di supporto: si pensi allo *Scrum Master* in Scrum, che ha il ruolo di facilitatore e di disseminatore della cultura Agile, non solo all'interno del team, ma nell'intero contesto aziendale. I ruoli tipici che si incontrano sono:

- *il Product Owner*, che si occupa di valorizzare quanto realizzato in funzione degli obiettivi di Valore richiesti dal cliente;
- *lo Scrum Master*, che si occupa di disseminare la Cultura Agile/Lean all'interno dello specifico team e del contesto aziendale in generale;

² Agile Manifesto, agilemanifesto.org



- *i developer*, a cui spetta il compito di realizzare fattivamente il progetto.

2.4 L'aspetto cloud

Al classico dualismo Hardware/Software, un progetto IoT aggiunge anche la specificità della **Componente Cloud**. Non si tratta, come spesso erroneamente si è portati a credere, di *“pubblicare i propri servizi ed il proprio database su una macchina virtuale remota”*, ma di abbracciare nuovi paradigmi di sviluppo in grado di rendere disponibili le proprie soluzioni come se fossero una commodity domestica: ne ho bisogno? li uso e pago; non ne ho bisogno? li spengo e non pago!

Per raggiungere tale obiettivo è fondamentale “pensare per il cloud” già in fase di realizzazione della soluzione: un esempio per tutti è quello della realizzazione di servizi “elastici” che “scalano” orizzontalmente, ovvero si aumenta la capacità di elaborazione complessiva aumentando il numero di macchine e bilanciando il carico tra di esse. Questo approccio architetturale è diametralmente opposto a quello che tipicamente viene seguito su soluzioni tradizionali in cui si cerca di “scalare” verticalmente, aumentando la potenza della singola macchina.

Discorso analogo per l'aspetto *Big Data* che deve essere gestito con estrema efficienza e con soluzioni moderne (si pensi allo storage in repository NoSQL), in modo da rendere i dati prontamente disponibili e “trasformabili” prima di essere “consumati”.

Lo sviluppo della Componente Cloud può essere affrontato tramite metodologie Agili, ma è necessario tener conto degli aspetti relativi alla piattaforma (o se si vuole, degli aspetti sistemistici), motivo per cui spesso si parla di *DevOps*, sia come metodologia che come ruolo a metà tra sviluppatore e sistemista.

2.5 La complessità della governance

Da quanto descritto deriva che la **complessità della governance** di un progetto IoT è un elemento tutt'altro che banale, e che richiede strumenti adeguati per armonizzare le attività ed il flusso operativo dei vari attori coinvolti.

Per fare ciò non è possibile prescindere dall'abbattimento dei muri che separano le logiche di sviluppo dei singoli componenti portanti, creando un processo uniforme che si basa sulla condivisione giornaliera dello stato di sviluppo e su una integrazione costante e veloce, finalizzata a rilasciare in maniera evolutiva ed incrementale una soluzione realmente utilizzabile ed ottenere rapidamente feedback.

Tutto ciò evidenzia come non sia realisticamente possibile affrontare la governance e lo sviluppo di una nuova soluzione IoT prendendo “in prestito” metodologie del mondo software, hardware o del Cloud, ma bisogna affrontare il problema in modo olistico.



AgileIoT

www.agileconstellation.info

In tale ottica, il mondo dello sviluppo software e quello dello sviluppo hardware si fondono in modo dirompente, creando un contesto nuovo con regole specifiche e meccanismi che ne sottendono la specifica realtà produttiva. Basti pensare alla terminologia utilizzata, spesso sottovalutata, ma fondamentale per il rapporto con il committente e i vari stakeholder coinvolti.



3. AgileIoT

Da queste considerazioni nasce **AgileIoT**, che, partendo dalla *Filosofia*, dai *Principi* e dalle *Pratiche* definite nell'**AgileConstellation Manifesto** ³, propone un approccio consistente e una metodologia per la creazione di Valore nel mondo dell'*Internet of Things*.

AgileIoT permette di non lasciare al caso la governance delle soluzioni che andranno a creare il nuovo ecosistema IoT, facendo tesoro dell'esperienza maturata fin ora nei vari ambiti disciplinari coinvolti.

Lo scopo è quello di non abbozzare degli adattamenti che potrebbero rivelarsi inadeguati, cercando di traslare aspetti specifici di un singolo dominio in un contesto in cui la multidisciplinarietà è un elemento portante.

Piuttosto, AgileIoT suggerisce di adottare un nuovo approccio che guarda all'Internet of Things in modo olistico, suggerendo di considerare sempre i diversi elementi interessati parte di un processo integrato.

3.1 AgileIoT: an AgileConstellation Star

AgileIoT è costituzionalmente basato sul mindset Agile e Lean, vantando la declinazione sviluppata in ambito del progetto AgileConstellation che guarda ai domini diversi da quelli software e digital in generale.

³ www.agileconstellation.it

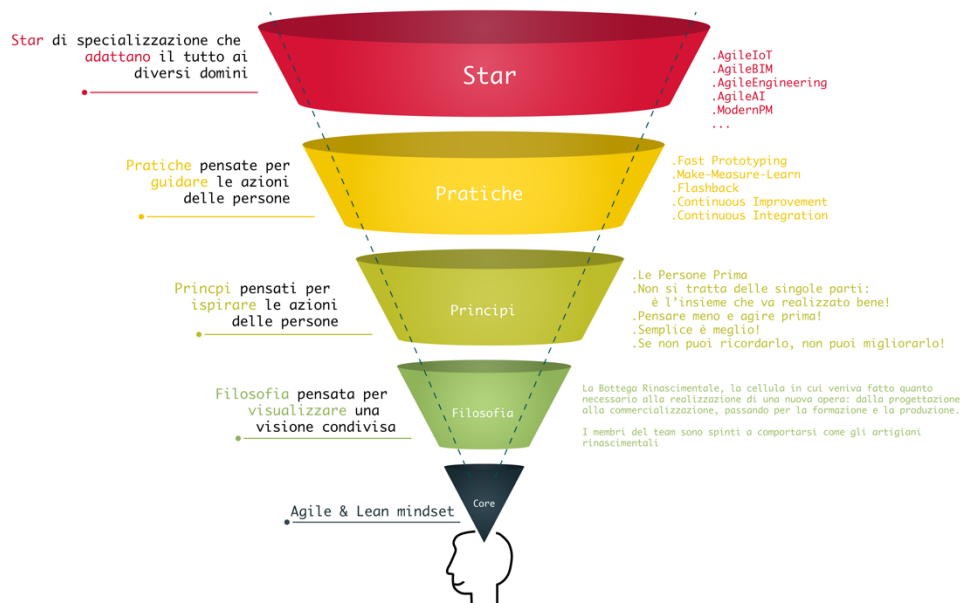




THE AGILECONSTELLATION FUNNEL

DISCOVER THE FOUNDATION

AgileConstellation.info



Version 2.1
The AgileConstellation Funnel & AgileConstellation by AgileConstellation.info is licensed under Creative Commons
Attribution-NonCommercial-NoDerivatives 4.0 International License.
Based on a work at www.agileconstellation.info.

Figura 3 - AgileConstellation funnel

AgileIoT eredita quindi quello che è il mindset, la filosofia, i principi (core) e le pratiche (core) di AgileConstellation, declinando ed estendendo, in particolare, questi ultimi due aspetti nel dominio specifico dell'IoT.

Abbiamo quindi:

- **Filosofia**, ispirata alla **Bottega Rinascimentale**, ovvero la cellula che assolve a quanto necessario per la realizzazione di una nuova opera: dalla progettazione, alla realizzazione e alla commercializzazione.
- **Principi (core):**
 - Non si tratta delle singole parti: è l'insieme che va realizzato bene!
 - Pensare meno e agire prima!
 - Semplice è meglio!
 - Se non puoi ricordarlo, non puoi migliorarlo!



- **Pratiche (core):**
 - *Fast Prototyping*, validare la sostenibilità della soluzione
 - *Make-Measure-Learn*, sperimentare rapidamente le diverse ipotesi e le diverse assunzioni
 - *Flashback*, azione di allineamento rapido in cui è l'osservatore ad ingaggiarsi nell'osservare l'elemento in lavorazione
 - *Continuous Improvement*, migliorare costantemente ogni aspetto
 - *Continuous Integration*, integrare costantemente le differenti anime della soluzione

3.2 AgileIoT Fast Prototyping

In accordo con l'approccio modulare dell'AgileConstellation Manifesto, la *Star AgileIoT*⁴ aggiunge 6 nuove bubble di dominio alla pratica di Fast Prototyping per validare la sostenibilità della soluzione:

- **Security:** incentrato sulla verifica delle ipotesi relative agli aspetti di security che caratterizzano l'intera soluzione e che ne influenzano in modo diretto lo sviluppo;
- **Energy:** incentrato sulla verifica delle ipotesi relative agli aspetti energetici in funzione delle esigenze di continuità operativa degli smart device;
- **Hardware:** incentrato sulla validazione delle ipotesi hardware tramite uno o più *Evaluation Kit* (EVK). La selezione dell'EVK più idoneo passa, in primis, dalla scelta della CPU/MCU, dopodiché si inizia a "costruire" il prototipo lavorando sugli altri componenti (es: RAM, USB, ecc.);
- **Code:** incentrato sulla prototipazione del firmware dei dispositivi e su quella dei servizi a supporto per l'acquisizione dei dati/eventi portanti della soluzione. In questa fase l'utilizzo di framework e di IDE di codifica veloce è fondamentale per abbattere i tempi relativi;
- **Data:** incentrato sugli aspetti legati alla raccolta, alla pulitura ed alla gestione dei *Raw Data* provenienti dai device, implementando gli aspetti di trasmissione e serializzazione degli stessi, con la scelta degli opportuni protocolli e dei formati più adeguati. In particolare, è fondamentale dedicare particolare attenzione alla stima dei volumi dei dati

⁴ www.agileconstellation.org



ed alle relative metodologie di analisi, in modo da approcciare alla gestione delle informazioni in chiave *Polyglot Persistence*, fondamentale per garantire la possibilità di elaborare velocemente grandi moli di informazioni;

- **Cloud:** incentrato sugli aspetti Cloud della soluzione, intesa come piattaforma di gestione dei dati, degli eventi e delle action portanti.



Figura 4 - Le nuove 6 bubble aggiunte dal dominio AgileIoT

Tutto il processo è guidato dal Product Owner (o dal Chief Product Owner in caso di più team) che, insieme al cliente, ai key stakeholder e al team, definisce la **Solution Big Picture** della soluzione e le **Epic** unitamente alla loro priorità.

È fondamentale evidenziare che tale fase impatta anche sulla strutturazione degli assett aziendali che devono supportare adeguatamente lo sviluppo della soluzione.

3.3 I Framework

È possibile implementare concretamente AgileIoT attraverso i due framework attuativi, **AgileIoT Duttile** e **AgileIoT Fiotto**.

Nel prosieguo andremo nel dettaglio dei due framework.

4. AgileIoT Duttile



AgileIoT Duttile definisce un processo **ricco ed articolato** per la produzione di soluzioni legate al mondo dell'Internet of Things, orientato al *Valore* e alle soluzioni End-to-End.

In particolare, la creazione di una specifica soluzione passa attraverso tre fasi ben delineate:

- **Prototyping Phase** (*timing: potenzialmente 2-4 settimane*): è la prima fase del processo. Viene definito il Vision Statement, effettuata la fase di Prototipazione Veloce e creato il Product Backlog attraverso una fase di planning specifica;
- **Engineering Phase** (*timing: quanto necessario al raggiungimento del Valore*): è la fase in cui la soluzione viene Ingegnerizzata e Sviluppata. Si tratta, intuitivamente, della fase più corposa e più complessa dell'intero processo;
- **Workout Phase** (*timing: potenzialmente 1-2 settimane*): è l'ultima fase focalizzata sul Delivery in esercizio, sul Supporto e sul Miglioramento Continuo.

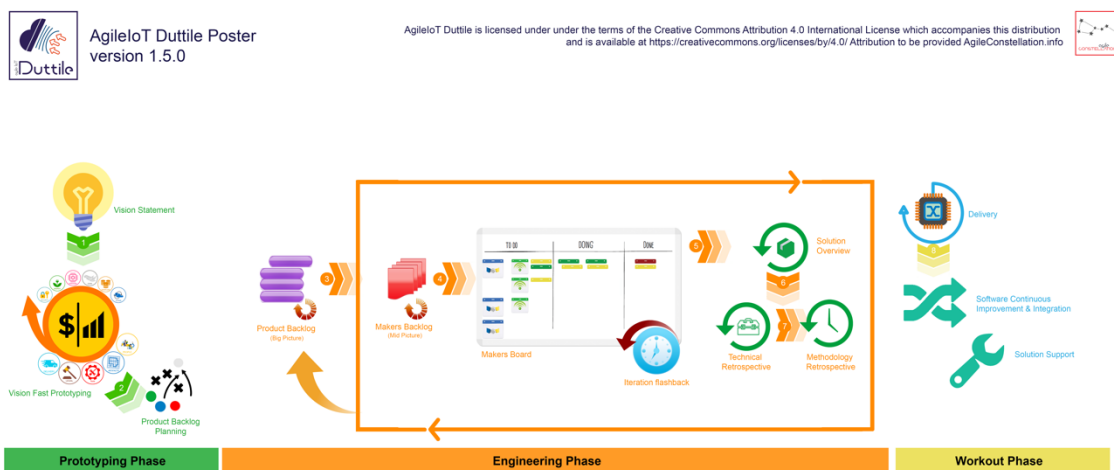


Figura 5 - AgileIoT Duttile Poster



La complessità intrinseca affida un ruolo fondamentale alla **Solution Definition of Done (sDoD)**, chiarendo in modo esplicito quando i seguenti Goal, contemplati da AgileIoT Duttile, possono ritenersi raggiunti.



Figura 6 – AgileIoT Duttile Goal

In particolare, oltre ai Goal delle singole fasi che si andranno a scoprire a breve, AgileIoT Duttile contempla una serie di obiettivi aziendali di crescita e gestione, definiti **Cross Phases Goal**:

- **Team Improvement**, *il Working Team opera in maniera tale da migliorarsi costantemente, sia da un punto di vista tecnico che da un punto di vista metodologico;*
- **Company Agility**, *i processi aziendali in essere devono diventare flessibili e consentire di abbattere barriere dettate da aspetti burocratici;*
- **Company Improvement**, *lo sviluppo della soluzione deve portare all'azienda anche un arricchimento di know-how in abbinamento al miglioramento dei processi interni;*
- **Buffer Budget**, *l'importanza di velocizzare tutte le fasi, soprattutto in funzione dell'hardware, porta alla necessità di disporre di un budget per le "spese giornaliere". Si pensi all'acquisto di un piccolo resistore per l'EVK (Evaluation Kit) per il costo di pochi euro: per tale attività non è pensabile che si debbano attivare complesse procedure d'acquisto che portano a settimane d'attesa.*

4.1 Prototyping Phase

La **Prototyping Phase** è la fase in cui l'azienda prende atto della volontà di realizzare una nuova soluzione e, partendo dall'idea o da bisogno iniziale, definisce quella che è il **Vision Statement**. Si tratta di tracciare gli obiettivi specifici in funzione delle esigenze del committente, andando a definire i **Solution Goal** e le *metriche di validazione*.



AgileIoT

www.agileconstellation.info

Essendo una fase di “scoperta” e “validazione”, è fondamentale affrontare sia elementi legati alle funzionalità della soluzione che si intende realizzare, sia elementi organizzativi.

Per fare ciò è necessario validare le **ipotesi** alla base della *Vision*, accertarsi che prodotto annesso è *realizzabile*, *strutturare i team e gli assett aziendali a supporto*, definire la **Solution Big Picture** e il **Product Backlog** di riferimento. Il tutto avviene sfruttando la pratica del *make-measure-learn*.

In dettaglio, per la *prototyping phase*, si hanno i seguenti Goal:

- **Vision Statement Definition**, viene definita la *Vision* della soluzione, identificando i bisogni da soddisfare e il mercato di riferimento;
- **Fast Prototyping Team**, viene creato il *Fast Prototyping Team*, acquisendo dall'esterno le eventuali figure necessarie;
- **Full Working Team**, viene formato il *Working Team*. In caso di più team, vengono definiti anche i gruppi di coordinamento ed i relativi coordinatori (*Chief*);
- **Solution Definition of Done**, viene definito il documento iniziale di riferimento per la *Definition of Done*;
- **Similarity Search**, in funzione della nostra *Vision*, e della soluzione derivante, viene sondato il mercato per verificare se esistono o meno proposte equivalenti. In caso affermativo, si procede ad un confronto qualitativo per capire come posizionarci nel mercato e distinguerci dai competitor. Si tratta di capire se siamo in un ambito *Red Ocean* (più player e regole già scritte) o *Blue Ocean* (primo player e regole da scrivere);
- **Hypothesis Validation**, vengono validate le ipotesi tramite l'evento di *Fast Prototyping*;
- **Set Work Environment**, viene predisposto, o al limite creata, l'area fisica di lavoro. Questo aspetto è particolarmente rilevante per lo sviluppo di una soluzione IoT perché la necessità di prototipare hardware richiede appositi spazi attrezzati: si pensi, ad esempio, alla qualità dell'aria in caso si effettui la saldatura di componenti elettronici;
- **Define Product Backlog**, viene definito il *Product Backlog* in funzione ai risultati della prototipazione.

Operativamente, la governance è affidata al *Fast Prototyping Temporary Team*, che partendo dal **Vision Statement** delinea gli scenari di utilizzo unitamente a una prima ipotesi architetture denominata *Intentional Architecture*. Le scelte vengono finalizzate e consolidate nella **Vision Fast Prototyping** (1) che porta al **Product Backlog Planning** (2).



4.2 Engineering Phase

Con l'**Engineering Phase** la realizzazione della soluzione entra nel vivo. In questa fase gran parte dell'enfasi è posta sullo sviluppo del prodotto, ma, parallelamente, anche gli aspetti a supporto vengono messi progressivamente a fuoco e strutturati a livello aziendale.

Il tutto parte con la definizione della **Mid Picture** (3) da parte del team (in caso di più team, ognuno di essi definisce la propria Mid Picture), ovvero con la definizione dell'obiettivo di iterazione in relazione ai Goal di riferimento:

- **Full Deployable Solution**, l'obiettivo è quello di creare una soluzione "chiavi in mano" per il cliente;
- **Reduce Uncertainty**, ridurre al minimo gli elementi di incertezza, in particolare quelli legati all'hardware e al firmware degli Smart Thing, ai servizi e agli aspetti Cloud;
- **Bill of Materials (BOM)**, viene definita la Smart Thing BOM;
- **Verify Full Solution**, stabilire e validare elementi oggettivi per valutare l'effettiva aderenza alle specifiche attese dagli stakeholder;
- **Testing Strategy**, individuare ed attuare opportune strategie di testing, funzionale e qualitativo;
- **Coordinate External Manufacturing**, gestire il rapporto con il team esterno di manufacturing incaricato di realizzare gli Smart Thing;
- **Packaging Verify**, attuare azioni di convalida del packaging degli Smart Thing e della soluzione in genere.

La Mid Picture si basa su un set attinente di Epiche da cui vengono definite le **Makers Story** durante il **Makers Backlog Planning**, tenuto sotto la governance del Product Owner. I **maker** definiscono i **Task** delle singole Makers Story e collocano il tutto sulla **Makers Board** (4).

La Makers Story composto dal minor numero di Task definisce il **Rhythm** della Mid Picture, che durante le fasi di realizzazione detta la cadenza del **flashback**.

Al completamento di tutti le Makers Story, e quindi di tutti i Task, viene effettuata la **Solution Overview** (5), presentando il prodotto funzionante al cliente, sotto la vigile organizzazione del *Product Owner*.

Nel caso di più team, la Solution Overview può essere anche ritardata in attesa dell'integrazione degli sviluppi dei singoli team.

Successivamente (6) viene eseguita la **Technical Retrospective** per discutere delle scelte tecnologiche, decidendo se *perseverarle* o fare *pivot*, seguita (7) dalla **Methodology Retrospective** per discutere sulla metodologia e l'organizzazione del team.



Prima di iniziare una nuova iterazione, si procede con il *grooming* del Product Backlog sotto la vigile guida del (Chief) Product Owner.

Un aspetto particolarmente delicato è quello della *creazione della componente hardware* degli Smart Thing, con l'EVK (che può essere diverso da quello della fase precedente) che cede gradualmente il posto all'hardware finale.

Possiamo individuare tre scenari tipo:

- *Custom Hardware;*
- *Market Hardware;*
- *Hybrid Hardware, basato su un mix tra Market e Custom.*

Il primo scenario è il più comune. Tutto viene realizzato specificamente per la soluzione ed evolve durante la fase di engineering. Tipicamente già nella prima iterazione si comincia a lavorare alla definizione della *Smart Things Bill of Materials* (BOM) utilizzando gli EVK finché non sono pronti i primi esemplari specifici e per tutta la fase di assestamento. Questo poiché i prototipi, soprattutto i primi esemplari, spesso presentano diversi problemi che vengono alla luce solo quando viene montato su il software di controllo.

Il secondo scenario è quello più raro. Si prende una soluzione di mercato e la si correda del software specifico. In tal caso si riduce drasticamente l'efficienza del tutto e l'intera attività di realizzazione della Soluzione diventa software-centric, non necessitando della progettazione hardware.

Il terzo scenario è sempre più interessante. Il CORE dell'hardware (CPU e Memoria) viene fornito già assemblato su specifica schedina, evitando di doversi districare tra i relativi problemi di progettazione/realizzazione, come, ad esempio, la tolleranza a disturbi e la vicinanza delle piste. Il resto viene progettato e realizzato specificamente in funzione delle esigenze della soluzione. In questo caso il ciclo di sviluppo segue quello della soluzione *custom*.

Nei due scenari più comuni, i team si occuperanno di produrre lo *Smart Thing BOM* che evolverà durante tutta la fase di engineering, grazie ad una apposita suite di test realizzata specificamente per verificarne il funzionamento, testando sia il firmware che il software.

La suite di test dovrà essere realizzata anche nel caso *Market Hardware*, anche se sarà chiaramente di diversa tipologia, essendo orientata alla verifica dei parametri funzionali e qualitativi di riferimento.

Prima di procedere è fondamentale chiarire l'aspetto legato alla **produzione manifatturiera** degli Smart Thing nel caso in cui non si scelga una soluzione di mercato.



Sia che la produzione avvenga da parte di un produttore terzo, che da parte di un diverso team/dipartimento della stessa azienda, dal punto di vista di AgileIoT Duttile essa è un'attività esterna. Come tale, va comunque correttamente gestita, in modo da allinearne lo sviluppo in funzione degli obiettivi di deployment. La gestione operativa è demandata al *Prime Maker* (o al *Chief Prime Maker* in caso di più team), con il supporto del *Product Owner* (o al *Chief Product Owner* in caso di più team) per gli aspetti amministrativi. Ad entrambi è affidato il compito di instaurare un proficuo rapporto di collaborazione e fiducia tra le diverse realtà produttive.

In particolare, il (Chief) Prime Maker, dovrà lavorare a stretto contatto con il team di sviluppo dell'hardware, al fine di chiarire i dubbi relativi in funzione della BOM e per "incastrare" correttamente i tempi relativi nel processo di engineering.

Una volta raggiunto il livello di stabilità desiderato, definito e comprovato dai test automatici relativi, la *Smart Thing BOM* assume la forma finale e non dovrebbe più essere necessario effettuare ulteriori modifiche, anche perché i costi relativi non sarebbero sostenibili, soprattutto a produzione avviata. Anche qui il (Chief) Prime Maker dovrà gestire la fase di sviluppo degli Smart Thing nella versione definitiva.

4.3 Workout Phase

L'ultima fase che interessa la soluzione è la **Workout Phase**, in cui il prodotto viene messo in esercizio (**deployment**) e vengono attivati i servizi a supporto del suo funzionamento:

- **Deployment Strategy**, viene definita la strategia di dispiegamento in produzione della soluzione nel suo insieme;
- **Solution Continuous Improvement**, dopo il rilascio, la soluzione viene costantemente migliorata, principalmente con continui aggiornamenti del firmware, dei servizi e degli aspetti Cloud;
- **Remote Improvement**, attuare soluzioni tecniche e tecnologiche che consentono di aggiornare quanto possibile direttamente da remote;
- **Solution Support**, viene prodotto tutto quanto necessario a guidare l'utilizzo della soluzione (ad esempio: manualistica utente) e attivato, eventualmente, un customer-care a supporto;
- **Retreat Strategy**, definire le opportune azioni di dismissione della soluzione o dell'eventuale ritiro in caso di gravi malfunzionamenti.

Una volta che si è a regime, si procederà a (8) supportare il cliente e migliorare costantemente la soluzione, intervenendo principalmente sul lato firmware, servizi e Cloud con le pratiche di **continuous improvement & integration**. Gli elementi hardware vengono modificati raramente



(anche a causa degli alti costi di intervento), a meno di un upgrade completo ad una nuova release della soluzione. È indispensabile contemplare strumenti che consentano il **deploy da remoto** degli aggiornamenti del firmware, soprattutto in quei contesti in cui il numero di Smart Thing è estremamente elevato e/o dislocato su una vasta area.

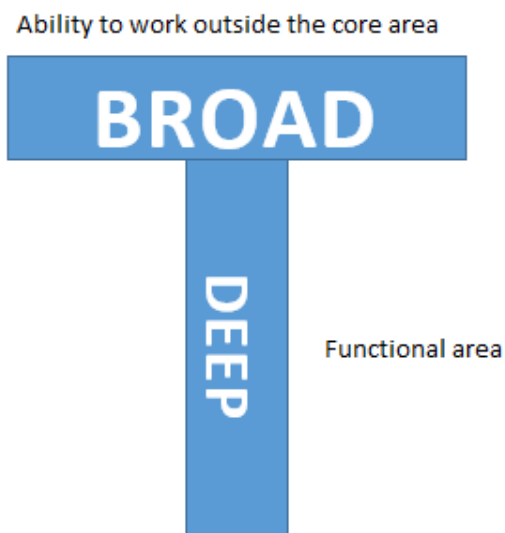
Bisogna tener presente che la soluzione ha, a differenza di quelle esclusivamente software o hardware, **due punti di interazione (2 point interaction)** con il cliente finale:

- **cloud endpoint;**
- **hardware endpoint.**

In pratica, il cliente dovrà districarsi sia con gli Smart Thing che con gli aspetti di analisi legati agli strumenti Cloud, cosa che richiede un supporto efficiente, passando attraverso azioni di formazione on-site, workshop, ecc.

4.4 Working Team

AgileIoT Duttile contempla Team che sposano la filosofia **T-Shaped**: i singoli membri hanno spesso competenze verticali profonde in un settore (*Deep*), ma sono in grado di intervenire su tutte le attività (*Broad*).



Il **Working Team** è composto da un numero variabile di **4:8 membri** afferenti ai *Primary Roles*:

- *T-Shaped Maker* :: 2, 4, 6
- *Prime Maker* :: 1
- *Product Owner* :: 1



a cui è possibile aggiungere ulteriori **5** figure inquadrabili come *Closer Roles*:

- *Cloud Closer* :: 1
- *Data Flow Closer* :: 1
- *Security Closer* :: 1
- *Energy Closer* :: 1
- *Legal Closer* :: 1

per un totale di **8:13** membri totali.

La distinzione tra *primary* e *closer roles* è funzionale al loro commitment sulla soluzione, con i primi impegnati per tutta la durata dello sviluppo ed i secondi presenti solo laddove sia necessario rinforzare il know-how specifico all'interno del team con azioni di coaching concentrate nella fase di prototyping e a supporto di quella di engineering.

Il suggerimento è di avere sempre coperti entrambi i ruoli in modo da garantire la realizzazione di una soluzione di alta qualità.

4.4.1 Primary Roles

T-Shaped Makers



I **T-Shaped Maker** (o anche semplicemente **maker**) sono i *maker* impegnati nella realizzazione delle *Makers Story*, ovvero dell'unità di lavoro atomica affidata al team e descritta in dettaglio successivamente.

Ogni Working Team contempla un numero *pari di maker* caratterizzati da 2 tipologie di skill primari: *Software Makers Oriented* e *Hardware Makers Oriented*.

Avere skill che consentono di coprire gli aspetti relativi sia alla "I" che alla "T" di una soluzione Internet of Things, permette al team di affrontare al meglio l'ampio spettro di problematiche che possono presentarsi durante la realizzazione della soluzione. Il confine non è, chiaramente, marcato in modo netto, permettendo una continua condivisione del know-how ed, eventualmente, uno switch (tipicamente temporaneo) da un domino all'altro.

Ai maker spetta la governance della *Makers Board*.

Prime Maker



Il **Prime Maker (PM, anche Prime)** è il *Team Lead Metodologico e Tecnologico* del Working Team, che indossa alternativamente il “cappello” più adatto in funzione delle esigenze temporalmente localizzate.

Il Prime aiuta il team nell’applicazione della metodologia, dedicandosi anche agli aspetti tecnici di alto livello, assumendo il ruolo di Solution Architect quando necessario e non disdegnando di sporcarsi direttamente le mani, andando a sostituire un Maker in caso di necessità, ma solo su esplicita richiesta degli altri Maker. In tal caso il Prime Maker ha il compito di creare una soluzione di riferimento, in modo da fornire nuovi strumenti e tecniche all’intero gruppo; si tratta, in sostanza, di comportarsi da *evangelist tecnologico*.

Nel caso siano presenti più Prime, perché in presenza di più team, è opportuno che venga costituita una *Prime Maker Board*, per allineare tra loro tutti i Prime, andando inoltre ad identificare il *Chief Prime Maker*.

Il Prime Maker (o, se presente, il *Chief Prime Maker*) si occupa di gestire le attività di approvvigionamento dell’hardware relativo agli Evaluation Kit ed ai relativi componenti. Inoltre segue direttamente la realizzazione esterna dell’hardware prototipale e finale, definendo le milestone e i processi di integrazione.

Product Owner



Il **Product Owner (PO)** è la **voce** degli stakeholder verso il team e la **voce** del team verso gli stakeholder. Il suo compito primario è quello di massimizzare il Valore della soluzione prodotta dal team in funzione di quelle che sono le necessità del cliente e dei key stakeholder. Nel caso siano presenti più Product Owner, perché in presenza di più team, è opportuno che venga costituita una *Product Owner Board*, per allineare tra loro tutti i PO, andando inoltre ad identificare il *Chief Product Owner*.

Il PO (o il *Chief PO coadiuvato dagli altri PO*) ha la governance esclusiva del *Product Backlog* ed è il responsabile delle relative attività di grooming. Allo stesso modo lo specifico PO (o l’unico) ha la governance esclusiva del *Makers Backlog*. Il (Chief) PO partecipa attivamente alla Solution Overview, mentre può essere coinvolto, laddove il team lo reputi necessario, in tutte le altre fasi del processo.

Non da ultimo, il (Chief) PO supporta, da un punto di vista amministrativo, la fase di realizzazione fisica degli Smart Thing, la quale, si ricorda, è operativamente seguita dal (Chief) Prime Maker.



4.4.2 Closer Roles

Energy Closer



L'**Energy Closer** è l'esperto degli aspetti energetici a cui sottende l'*operatività dello Smart Thing*, impattando sulla sua capacità di garantire una continuità sufficiente ad assolvere le proprie mansioni nell'intervallo di tempo previsto. Diventano quindi particolarmente importanti scelte come quelle legate al tipo di *accumulatore* da utilizzare e la *fonte di ricarica*, soprattutto se gli Smart Thing devono operare in zone in cui non è possibile usufruire della rete elettrica, rendendo indispensabile ricorrere a energie alternative come quella solare.

Il compito dell'Energy Closer è quello di tenere costantemente il team focalizzato sugli aspetti energetici nella loro completezza.

Data Flow Closer



Il **Data Flow Closer** supporta il team nelle fasi di analisi delle tecnologie e degli strumenti necessari per la "**raccolta**", la "**pulizia**" e la "**gestione**" dei *Raw Data* provenienti dai dispositivi. Partendo dal dato grezzo in uscita dal dispositivo, è necessario scegliere i protocolli di trasferimento dello stesso, la relativa modalità di serializzazione e clean up, e la tecnologia di persistenza più indicata in chiave *Polyglot Persistence*.

Cloud Closer



Il **Cloud Closer** è lo specialist a supporto delle attività di analisi dei dati/eventi e sulla generazione di opportune rappresentazioni e action di risposta. Siamo nel campo dei **Big Data** e degli **Event Driven Action**, ed è fondamentale avere opportune soluzioni **Cloud** (la "I") che permettano di gestire tali elementi in modo efficace ed efficiente.

Il compito del Cloud Closer è quello di tenere costantemente il team focalizzato sugli aspetti legati al Cloud.

Security Closer



Il **Security Closer** da supporto sugli aspetti di *Security* della soluzione, particolarmente rilevanti vista l'importanza dei dati, degli eventi e delle relative azioni che ne scaturiscono e dalla loro declinazione in funzione del contesto specifico.



Si tratta di un ambito particolarmente importante, essendo l'IoT è un campo totalmente nuovo che rischia di prestare il fianco a numerosi problemi di affidabilità, efficienza e sicurezza, amplificati dall'uso *headless & connected*.

Il compito del Security Closer è quello di tenere costantemente il team focalizzato sugli aspetti legati alla *security*.

Legal Closer



Il **Legal Closer** è uno specialista non tecnico, ma in grado di supportare il team nell'analisi di quelle che sono le normative da rispettare sia nella realizzazione della Soluzione nel suo complesso, sia nella gestione successiva alla sua creazione. Si pensi, come esempio, alle norme che prevedono l'utilizzo di determinate frequenze wireless o l'adozione di specifiche misure di sicurezza, sia del sistema nel suo insieme, che nella gestione dei dati. Nell'ambito di soluzioni IoT lo scenario è ulteriormente complicato dal contesto internazionale a cui queste si rivolgono, nel quale esistono notevoli differenze tra le normative applicabili.

A differenza degli altri *Closer*, il Legal Closer svolgerà un'intensa attività nella fase di prototyping, per poi accompagnare, in qualità di consulente, il Team durante tutte le fasi successive.

4.4.3 Temporary Team

In funzione delle specifiche fasi, i membri del Working Team danno vita ai **Team Temporanei (T2)** che assolvono a specifiche esigenze di contesto. L'outcome delle relative attività viene successivamente condiviso con il resto dei membri durante il planning e le retrospettive.

Fast Prototyping Temporary Team



Il **Fast Prototyping Temporary Team (FPT2)** è il team temporaneo che si occupa del Fast Prototyping previsto nella fase di Prototype. Tipicamente è costituito dal Prime Maker, da due T-Shaped Maker e, possibilmente, dai *Closer*, andando a rappresentare il nucleo del futuro **Working Team**.

Story Temporary Team



Lo **Story Temporary Team (ST2)** è il *team temporaneo* a cui è affidato lo sviluppo della singola Makers Story.

Gli ST2 possono essere *Cloud Oriented* e *Smart Thing Oriented*, in funzione dell'attività da



svolgere, andando a coprire sia la “I” che la “T”, e sono sempre composti da due maker con skill differenti. Si tratta di una nuova interpretazione della pratica del *pair-programming* (o non-solo programming), in cui l’ST2 diventa la cellula di lavoro fondamentale e portante dell’intero processo di realizzazione.

4.5 Artefatti

Gli **Artefatti** sono gli elementi che caratterizzano il processo e che consentono di concentrare l’attenzione e gli sforzi del team, al fine di descrivere lo stato di avanzamento di realizzazione della soluzione.

Ogni artefatto ha una governance specifica che appartiene ad uno o più membri del Working Team ed è soggetto a continue azioni di review al fine di allinearli allo stato corrente del progetto ed al know-how maturato.

Work Item

Epic



Una **Epica** enfatizza il *Valore* in funzione del cliente, cogliendo aspetti trasversali che guidano e condizionano lo sviluppo.

Per chiarire il concetto facciamo un esempio: si immagini di progettare una soluzione che monitori il traffico in una determinata strada. Per fare ciò si andranno ad osservare diversi fattori come: *il numero di auto in transito nell’unità di tempo, la qualità dell’aria, il corretto funzionamento dei semafori, ecc.* Ognuno di tali fattori è appunto una Epica.

Le Epiche sono accompagnate dai *Criteri di Accettazione*, ovvero una esplicita definizione del comportamento atteso nelle condizioni di esercizio previste. Essi impattano sia sugli aspetti tecnici, ad esempio test funzionali, che su quelli di supporto, ad esempio l’aggiunta delle relative specifiche al datasheet.

Makers Story



La **Makers Story** (o semplicemente Story) è l’elemento su cui si concentra l’attività dei Maker, in particolare dell’ST2. Si tratta di un’attività descrivibile in funzione di elementi di input/output dei dispositivi IoT specifici, suddivisibile in due categorie, sempre in funzione della “T” e della “I”:

- **Device Makers Story**, ovvero le Makers Story specificamente orientante allo sviluppo degli Smart Thing, che possono essere identificate in:
 - *Measure Story*, sono le Makers Story che hanno come scopo primario quello di misurare parametri esterni come sensori ambientali. In questo caso l'input è il dato del sensore e l'output è il valore letto, eventualmente normalizzato;
 - *Act Makers Story*, sono le Makers Story caratterizzate da attuatori che effettuano un'azione in funzione dei parametri rilevati. In questo caso l'input è il dato del sensore di controllo e l'output è l'esito dell'azione/correzione effettuata in funzione di esso.
- **Cloud Makers Story**, sono le Makers Story focalizzate sugli aspetti di elaborazione/analisi/processing in Cloud:
 - *Data Makers Story*, sono le Makers Story legate all'elaborazioni di dati, che rappresentano l'elemento principale del computo stesso;
 - *Event Makers Story*, sono le Makers Story legate ad un evento/allarme che non hanno focus su dati specifici.

Restando sull'esempio del monitoraggio del traffico stradale, prendiamo l'Epica relativa al numero di auto in transito nell'unità di tempo. Rispetto ad essa potremmo ipotizzare di avere le seguenti Makers Story:

- *Smart Thing che conta il numero di veicoli in transito (device story);*
- *Invio dei dati al sistema di raccolta e gestione (device story);*
- *Analisi e gestione dei dati in formato aggregato (cloud story);*

Essendo le Makers Story rappresentative di una esigenza di business, tale distinzione può essere ritenuta anche solo letteraria, non evidenziandola direttamente, come invece suggerito di seguito per i Task.

Task

Un **Task** è l'unità minima di lavoro, possibilmente specializzata per singola area applicativa: *hardware [task]*, *firmware [task]*, *service [task]* e *cloud [task]*. Per ottenere una chiara rappresentazione visiva dell'area afferente, è importante utilizzare colori differenti per rappresentare i Task stessi.

Specificare la tipologia di Task è molto utile per gestire le diverse specificità: ad esempio, fare una modifica dell'hardware a progetto inoltrato è molto rischioso e costoso, fattore che potrebbe spingere a ragionare su una possibile risoluzione del problema basata sulla modifica del firmware. In questo caso si adotterebbe una soluzione probabilmente meno efficiente, ma riducendo il rischio di aumentare drasticamente i costi ed allungare i tempi.



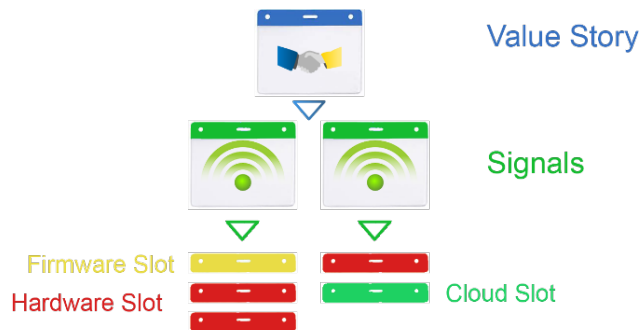


Figura 7 - Epic, Story e Task

4.6 Visual Management

Product Backlog



Il **Product Backlog** contiene le **Epiche** della soluzione, ordinate per priorità in funzione delle scelte del cliente. Si tratta di uno stack di elementi ordinati a priorità crescente in chiave top-down: l'elemento al top è quello a maggior valore relativo.

Il *Product Owner* (o al Chief PO coadiuvato dagli altri PO) ha la governance del Product Backlog e delle attività primarie ad esso collegate:

- *Product Backlog planning*, ovvero il planning dedicato alla Vision generale della soluzione. Durante il Product Backlog planning vengono create e dettagliate le Epiche da sviluppare in funzione delle necessità dei (key) stakeholder;
- *Product Backlog grooming*, ovvero l'attività di governance del Product Backlog, affidata al Product Owner (o al Chief PO coadiuvato dagli altri PO) che lo allinea dopo la Solution Review, in risposta ai feedback raccolti.

Makers Backlog



Il **Makers Backlog** contiene tutti gli elementi necessari alla realizzazione della soluzione, a cominciare dalle *Makers Story*, ordinate (in priorità) in funzione del Rischio, in modo esplicito, e del Valore, in modo implicito, perché legato alle Epiche già priorizzate in tal senso.

Il Makers Backlog è sotto la governance di uno specifico Product Owner, che, insieme ai maker, lo popola attraverso lo specifica evento di *Makers Backlog planning*, ovvero il planning dedicato alla successiva iterazione.

Makers Board



La **Makers Board** consente di orchestrare lo sviluppo delle Makers Story, e degli altri elementi di lavoro, da un punto di vista del processo. Di base sono previste tre colonne:

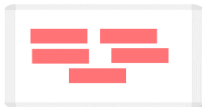
- *To Do*, contiene le Makers Story selezionati e i relativi task costituenti;
- *Doing*, contiene i Task attualmente in lavorazione;
- *Completed*, contiene i Task completati.

Ogni *Temporary Team* può essere al lavoro esclusivamente su una sola Makers Story ed avere nello stato “Doing” un solo Task dello stesso tipo. In tal modo si evita che l’ST2 sprechi tempo ed energia in attività *cross-Story* e *cross-Task* (multi tasking), concentrandosi su un obiettivo preciso e ben delineato.

La colonna “Doing” e la colonna “Completed” sono caratterizzate da specifiche *Definiton Of Done*, che, nel primo caso, riguardano fortemente gli *aspetti qualitativi* delle Makers Story, mentre nel secondo, quelli *operativi*, ovvero di utilizzo finale.

La Makers Board è sotto la governance dei maker.

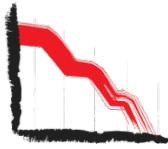
Technical Debt & Risk Board



La **Technical Debt & Risk Board** è uno strumento di supporto a tutti gli aspetti realizzativi della soluzione. Si tratta di un modo per tener traccia sia il *Debito Tecnico*, termine comunemente utilizzato per indicare le eventuali conseguenze derivanti da una frettolosa progettazione e realizzazione, sia dei *Rischi* annessi a scelte che vanno prontamente affrontati.

Il dato di fatto è che ogni progetto ha sempre una particolare percentuale di rischio, così come ha sempre un debito tecnico, involontario o volontario, che tendenzialmente cresce durante la sua realizzazione. Tracciarne gli aspetti caratterizzanti consente di intervenire costantemente su di essi per ridurre al minimo l’impatto sulla qualità finale della soluzione.

Burndown Chart



Il **Burndown Chart** consente di monitorare l'avanzamento delle attività correlando i Task (o le Makers Story) al fattore "tempo". Ciò permette di intervenire rapidamente per risolvere i problemi che portano a discostarsi dall'andamento ideale ipotizzato.

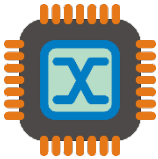
4.7 Delivery Item

Smart Thing Bill Of Materials



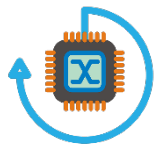
Lo **Smart Thing Bill Of Materials (BOM)** è l'elenco di tutti i componenti, sottoinsiemi, semilavorati e materie prime necessari per realizzare l'hardware.

Smart Thing



Lo **Smart Thing** è il componente intelligente alla base della soluzione IoT che si sta realizzando. La parte hardware può essere direttamente acquistata da un produttore, se esiste già sul mercato, o progettata e realizzata specificamente.

Solution to Delivery



La **Solution to Delivery (StD)** è la soluzione funzionante ed incrementale da mostrare al raggiungimento della *Mid Picture*, durante la *Solution Overview*.

In generale comprende:

- *uno o più Smart Thing;*
- *soluzioni Cloud a Supporto;*
- *componentistica di Delivery.*

L'StD deve essere in linea con *Solution Definition of Done*.

Solution Definition of Done Document



Il **Solution Definition of Done Document (sDoD Document)** è un *live-document* che definisce gli elementi chiave per considerare la soluzione completa. Si tratta di chiarire in modo univoco quando è possibile ritenere che la soluzione rispecchia gli obiettivi di Valore di riferimento ed è quindi possibile passare dalla fase di Engineering a quella di Workout.



La sDoD condiziona direttamente quelli che sono i *Criteri di Accettazione* legati alle *Epiche* e alle *Makers Story*, tenendo traccia anche dell'evoluzione delle specifiche della soluzione stessa, per cui, se un elemento non è più funzionale all'obiettivo, esso non viene cancellato, bensì marcato come obsoleto. In linea generale, lo sDoD Document si presenta come una short list, redatta in prima

battuta durante la fase di Prototyping.

4.8 Eventi

Gli **Eventi** sono precisi momenti in cui si concentrano specifiche azioni di pianificazione o di review.

Vision Fast Prototyping

Il **Vision Fast Prototyping**, basato sulla pratica omonima e declinata per la Star IoT⁵, è l'evento dedicato alla prototipazione veloce in funzione della Vision, validando la sostenibilità dell'iniziativa e ponendo le basi per il *Product Backlog planning*.

Applicando il loop Make-Measure-Learn le azioni sono immediate e finalizzate all'apprendimento delle caratteristiche principali annesse agli specifici elementi da analizzare e verificare.



Team.



Il *Fast Prototyping* dura potenzialmente dalle **2** alle **4 settimane**.

⁵ Si guardi il paper di AgileIoT su www.agileconstellation.info per i dettagli sulla stessa



Planning

Backlog Planning



AgileIoT Duttile contempla due Eventi primarie di planning: *Product Backlog*

planning e *Makers Backlog planning*.

Il *Product Backlog planning* ha lo scopo di “popolare” il Product Backlog in funzione della Big Picture ovvero delle esigenze di Valore dei (key) stakeholder. La sua definizione avviene a valle del fast prototyping. Dualmente, il *Makers Backlog planning* ha lo scopo di popolare il Makers Backlog con le Makers Story, relativi alle Epiche, e i Task afferenti funzionali al raggiungimento della Mid Picture.



Il *Product Backlog planning* è sotto la governance del Product Owner o del Chief Product Owner nel caso di più team. Dualmente, la governance del *Makers Backlog planning* è di pertinenza dello specifico, o del singolo, Product Owner



Il *Product Backlog planning* dura potenzialmente **16 ore** suddivise in **2 sessioni**, mentre il *Makers Backlog planning* **4 ore** in relazione ad un'iterazione tipo di 2 settimane (8 ore nel caso di 4 settimane).

Refinement



Il **refinement** è l'azione che si applica, singolarmente e in momenti distinti, al *Product Backlog* e il *Makers Backlog* in funzione dei feedback e dell'evoluzione della soluzione.



Il *Product Backlog grooming* è sotto la governance del Product Owner (o del Chief PO coadiuvato dagli altri PO) coadiuvato dall'intero team (o dai team). Il *Makers Backlog grooming* è sotto la governance dello specifico Product Owner coadiuvato dallo specifico team.



Rispetto ad un'iterazione tipo di 2 settimane, il *Product Backlog grooming* può durare potenzialmente dalle **2 alle 4 ore**, in funzione dei feedback raccolti. Il *Makers Backlog grooming* può durare potenzialmente al massimo **1 ora**.

Iteration Flashback



L'**Iteration Flashback** rappresenta il momento fondamentale di condivisione del know-how e dello stato delle attività durante le iterazioni. Basato sulla pratica di *flashback*, la sua cadenza è dettata dal *Rhythm*, ogni ST2 “visita” gli altri ST2 ponendo al massimo *2 domande mirate*.

Questo evento di **allineamento rapido** è pensata per il mondo specifico dell'IoT in cui la soluzione è composta da componenti hardware che non possono essere continuamente spostati, andando a sostituire il classico Stand Up meeting. In tal modo è l'*osservatore* ad andare direttamente al *desk* su cui presente l'attuale manufatto in lavorazione.

Dal flashback può emergere la necessità di approfondire alcuni aspetti, cosa che porta ad una *Technical Review Blitz* in cui dirimere ogni dubbio.



Il flashback è di pertinenza dei **makers**, con il supporto, laddove richiesto, del *Prime Maker*.



Ogni ST2 dedica **3 minuti** ad “osservare” il lavoro degli altri *temporary team*. La frequenza dei feedback è definita dal *Rhythm*.

Retrospective

Solution Overview



La **Solution Overview** viene attuata per mostrare l'STD al customer e agli stakeholder. Durante la Solution Overview il Product Owner (o il Chief PO coadiuvato dagli altri PO) raccoglie i feedback per il grooming del Product Backlog.

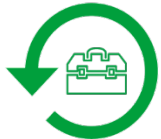


La solution overview è guidata dal Product Owner (o del Chief PO), assistito dal Prime Maker (o del Chief Prime Maker) e, dove necessario, dagli altri membri dal Working Team.



Rispetto ad un'iterazione tipo di 2 settimane, dura potenzialmente **3 ore**.

Technical Retrospective



La **Technical Retrospective** ha lo scopo di fare il punto sulle tecnologie utilizzate e sui relativi strumenti ed approcci di sviluppo, in funzione dei risultati ottenuti. L'obiettivo è quello di validare le scelte o di approcciarne altre laddove gli esiti non sono in linea con le attese: **preserve or pivot**.

Rispetto all'approccio classico dell'Agile, dove la retrospettiva è tipicamente solo di processo, AgileIoT Duttile contempla esplicitamente un evento per parlare degli aspetti tecnici soprattutto in relazione alle problematiche e soluzioni hardware.

Nel caso di più team, la Technical Retrospective viene tenuta singolarmente da ogni team e, successivamente, gli specifici Prime allineano gli altri durante un meeting della Prime Maker Board.



La *technical retrospective* è guidata dal *Prime Maker*, nel ruolo di facilitatore e vi partecipa l'intero Working Team.



Rispetto ad un'iterazione tipo di 2 settimane, dura potenzialmente **2 ore**. 1/4 del tempo dedicato agli aspetti legati ai dati.

Methodology Retrospective



La **Methodology Retrospective** viene eseguita per discutere sull'applicazione della metodologia e sull'organizzazione del team. Obiettivo: continuous improvement.



Nel caso di più team, la Methodology Retrospective viene tenuta singolarmente da ogni team e, successivamente, gli specifici Prime allineano gli altri durante un meeting della Prime Maker Board.



La methodology retrospective è guidata dal *Prime Maker* che ha il ruolo di facilitatore e vi partecipa l'intero team.



Rispetto ad un'iterazione tipo di 2 settimane, dura potenzialmente **3 ore**.

4.9 Metriche

Le **Metriche** consentono al Working Team di monitorare l'andamento delle attività e di pianificare le Eventi e le altre azioni a supporto di esse.

Rhythm



Il **Rhythm** definisce la cadenza di esecuzione del flashback. Viene calcolato in funzione della Makers Story con minor numero di Task associati:

- *Story di 1 - 2 Task: Daily Rhythm, flashback giornaliero;*
- *Story di 4 - 8 Task: b-Daily Rhythm, flashback bi-giornalieri;*
- *Story di 12+ Task: t-Daily Rhythm, flashback tri-giornalieri;*

Lead Time & Throughput



Il **Lead Time** è il tempo medio impiegato da una Makers Story per essere completato, dalla sua formalizzazione al suo dispiegamento.

Il **Throughput** è la stima del tempo necessario a completare il Makers Backlog, dato da:

- *Throughput = All Makers Story * Lead Time.*



5. AgileIoT Fiotto



AgileIoT Fiotto è la soluzione in chiave “*Continuous Value*” e “*Continuous Deployment*” per la realizzazione di soluzioni IoT secondo l’*AgileConstellation Manifesto*.

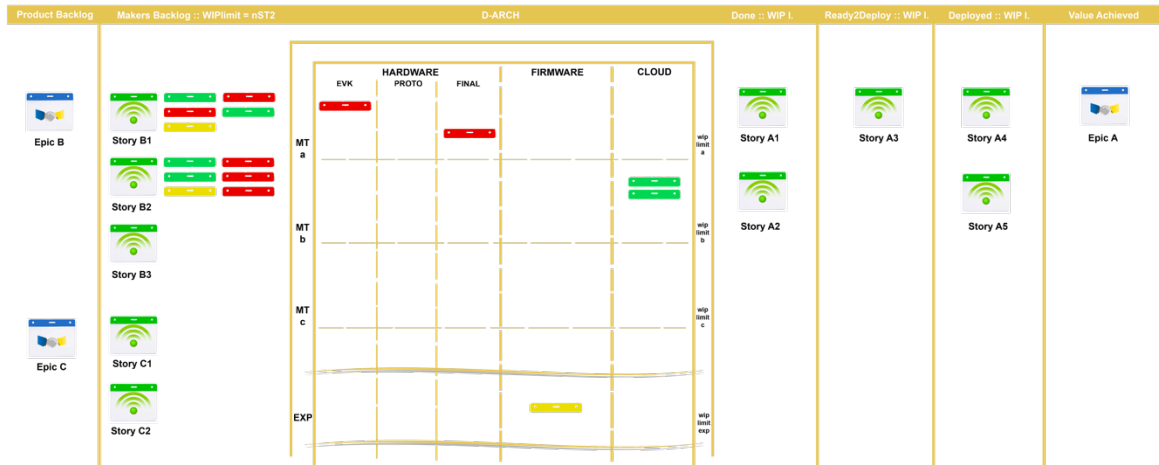


Figura 8 - AgileIoT Fiotto

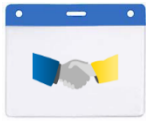
Il **WorkPivot** consente di passare dall’evidenza delle attività afferenti l’intero *Fiotto Team* (verticali) a quelle del singolo *Makers Team* (orizzontali), gestendo la trasformazione visivamente grazie all’introduzione del **D-ARCH** (*Do it for... Achieve Rapidly Customer Hopes*). Le diverse attività (Task) sono identificate in funzione della loro specificità (*Hardware*, *Firmware* e *Cloud* – asse verticale) associandole, contemporaneamente, al Makers Team (asse orizzontale) e al relativo *Work in Progress limit* (WIP-limit / WIP-I).

5.1 Work Item

I **Work Item** sono gli elementi di lavoro su cui si concentra il *Fiotto Team*, e i vari *Makers Team*, consentendo di descrivere implicitamente lo stato di avanzamento di realizzazione della soluzione.



Epic



Una **Epic** enfatizza il *Valore* in funzione del cliente, cogliendo aspetti trasversali che guidano e condizionano lo sviluppo.

Per chiarire il concetto facciamo un esempio: si immagini di progettare una soluzione che monitori il traffico in una determinata strada. Per fare ciò si andranno ad osservare diversi fattori come: *il numero di auto in transito nell'unità di tempo, la qualità dell'aria, il corretto funzionamento dei semafori, ecc.* Ognuno di tali fattori è appunto una Epic.

Makers Story



La **Makers Story** (o Story) è l'elemento su cui si concentra l'azione del Makers Team. Si tratta di un'attività descrivibile in funzione di elementi di input/output dei dispositivi IoT specifici, suddivisibile in due categorie:

- **Device Story**, specificamente orientanti allo sviluppo degli Smart Thing, che possono essere dettagliate in:
 - *Measure Story*, che hanno come scopo primario quello di misurare parametri esterni come sensori ambientali. In questo caso l'input è il dato del sensore e l'output è il valore letto, eventualmente normalizzato;
 - *Act Story*, caratterizzate da attuatori che effettuano un'azione in funzione dei parametri rilevati. In questo caso l'input è il dato del sensore di controllo e l'output è l'esito dell'azione/correzione effettuata in funzione di esso.
- **Cloud Story**, focalizzate sugli aspetti di elaborazione/analisi/processing in Cloud:
 - *Data Story*, legate all'elaborazione di dati. Rappresentano l'elemento principale del computo stesso;
 - *Event Story*, legate ad un evento/allarme che non hanno focus su dati specifici.

Restando sull'esempio del monitoraggio del traffico stradale, prendiamo la Epic relativa al numero di auto in transito nell'unità di tempo. Rispetto ad essa potremmo ipotizzare di avere le seguenti Makers Story:

- *Smart Thing che conta il numero di veicoli in transito (device story);*
- *Invio dei dati al sistema di raccolta e gestione (device story);*
- *Analisi e gestione dei dati in formato aggregato (cloud story);*



Essendo le Makers Story rappresentative di una esigenza di business, tale distinzione può essere ritenuta anche solo letteraria, non evidenziandola direttamente, come invece suggerito di seguito per i Task.

Un **Task** è l'unità minima di lavoro, possibilmente specializzata per singola area applicativa: *hardware [task]*, *firmware [task]*, *service [task]* e *cloud [task]*. Per ottenere una chiara rappresentazione visiva dell'area afferente, è importante utilizzare colori differenti per rappresentare i Task stessi.

The diagram illustrates the decomposition of an Epic into Makers Story, which then branches into Firmware Task, Hardware Task, and Cloud Task.

- Epic**: Represented by a blue icon with a blue and yellow arrow.
- Makers Story**: Represented by a green icon with a green and yellow arrow.
- Firmware Task**: Represented by a yellow icon with a yellow and red arrow.
- Hardware Task**: Represented by a red icon with a red and green arrow.
- Cloud Task**: Represented by a green icon with a green and blue arrow.

5.2 Fiotto Team

Nello specifico, il team si organizza in sotto gruppi dando vita ai **Makers Team (MT)**, ovvero ai *team temporanei* a cui è affidato lo sviluppo della singola Makers Story. Gli MT possono essere *Cloud Oriented e Smart Thing Oriented*, in funzione dell'attività da svolgere e sono sempre

composti da almeno due maker con skill differenti. Si tratta di una nuova interpretazione della pratica del *pair-programming* (o non-solo programming), in cui il Makers Team diventa la cellula di lavoro fondamentale e portante dell'intero processo di realizzazione.

5.3 Aree

Fiotto è suddiviso in 7 aree primarie di riferimento (colonne):

- **Product Backlog**, contiene le Epic del prodotto, ordinate in funzione del Valore relativo;
- **Makers Backlog**, contiene le Makers Story, e i relativi Task, che compongono le Epic pronte per essere realizzate. Quest'area ha, di base, un WIP Limit pari al numero di MT, spingendo i singoli team temporanei a concentrarsi su un unico obiettivo alla volta. Chiaramente tale limite può essere variato in funzione alle specifiche esigenze;
- **D-ARCH**, è l'area di lavorazione degli Task. Verrà descritta in dettaglio di seguito;
- **Done**, quando una Makers Story raggiunge quest'area, vuol dire che tutti i Task relativi sono stati completati e si può procedere a pianificare ed attuare tutte le azioni preparatorie al suo Deployment. Si possono, ad esempio, attivare i nuovi servizi del Customer Service e di Operation atti a supportare il nuovo elemento in produzione;
- **Ready to Deploy**, le Makers Story in quest'area devono essere solo dispiegate. Ad esempio, si procede a concordare con cliente la data effettiva di deploy;
- **Deployed**, la Makers Story è stata dispiegata ed è operativa;
- **Value Achieved**, quando una Epic raggiunge quest'area, vuol dire che tutte le Makers Story afferenti sono state dispiegate e la soluzione è operativa. È fondamentale che i criteri di accettazione previsti dalla *sDOD*, ovvero la *Solution Definition of Done*, siano stati soddisfatti.

5.4 WorkPivot

WorkPivot è lo strumento di trasformazione di Fiotto che consente di passare dalle attività complessive della soluzione, e quindi di interesse per l'intero Fiotto Team, a quelli più legati all'attività del singolo Makers Team.

Fiotto si presenta con entrambe le caratteristiche, passando in modo "elegante" da una visione all'altra grazie all'introduzione del D-ARCH.



AgileIoT

www.agileconstellation.info

5.5 D-ARCH

Il **D-ARCH** è il cuore pulsante di Fiotto, evidenziando le attività in essere e gli elementi “pull” che ne cadenzano il ritmo.

D-ARCH è un acronimo per: **Do it for... ACHIEVE RAPIDLY CUSTOMER HOPES**, ovvero “fallo per.. centrare velocemente le aspettative del cliente”.

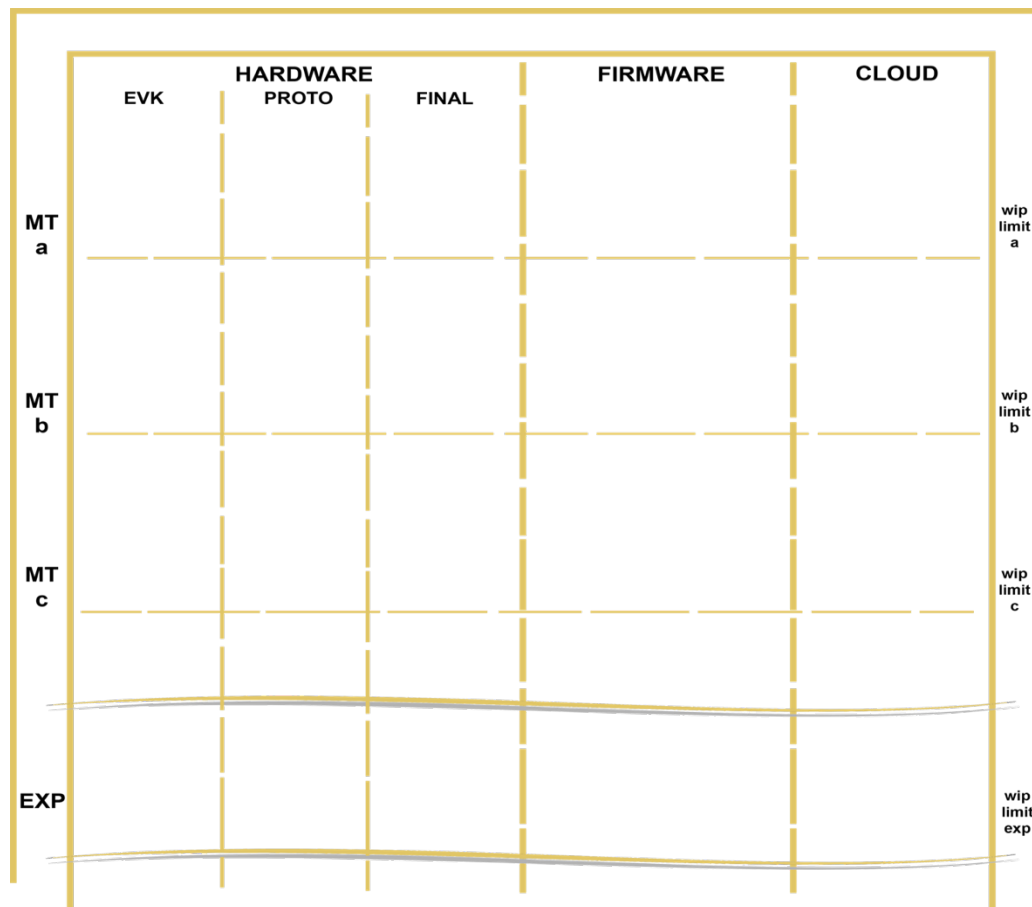


Figura 10 – D-ARCH

Il D-ARCH capovolge la vista d’insieme (sfruttando il *WorkPivot*), andando a spostare il focus sull’attività dei singoli MT e creando una specifica caratterizzazione delle dimensioni *Verticali* ed *Orizzontali*:

- *Dimensione Verticale*: evidenzia gli aspetti afferenti all’intero Fiotto Team e al progetto in generale;
- *Dimensione Orizzontale*: evidenzia gli aspetti afferenti ai singoli Makers Team.



I pilastri dell'arco identificano lo specifico *Makers Team* e il relativo *WIP limit*, mentre lo spazio interno è suddiviso in colonne che identificano la singola tipologia di *Task*: **hardware**, **firmware** e **cloud**.

In particolare, l'area dedicata ai Task Hardware è a sua volta suddivisa in: **EVK**, **Smart Thing Proto** e **Smart Thing Final**, volendo evidenziare in modo esplicito le differenti fasi di realizzazione dell'hardware annesso allo Smart Thing.

Dopo le sub-aree dei Makers Team, è presente una sub-area dedicata agli **Esperimenti (EXP)**, dove è possibile collocare Task utili per verificare ipotesi, valutare rischi ed acquisire nuovo know-how specifico.

La sub-area di sperimentazione, delimitata da due swimlane, è volutamente indicata come dimensione orizzontale e non associata verticalmente ai singoli Makers Team. Questo per non limitare la sperimentazione al team specifico ma spingere anche alla formazione di **Experimental Team (ET)** temporanei, sempre costituiti da una coppia di Makers, supportando la condivisione del know-how ottenuto dagli esperimenti.

5.6 D-ARCH Metriche

Un aspetto estremamente interessante del **D-ARCH** è la possibilità di associare diverse metriche sia alla dimensione verticale che orizzontale, ottenendo degli indicatori molto utili per le azioni di *Continuous Improvement* del Fiotto Team e nella gestione del progetto.

Le principali metriche individuate sono riportati nella figura seguente:



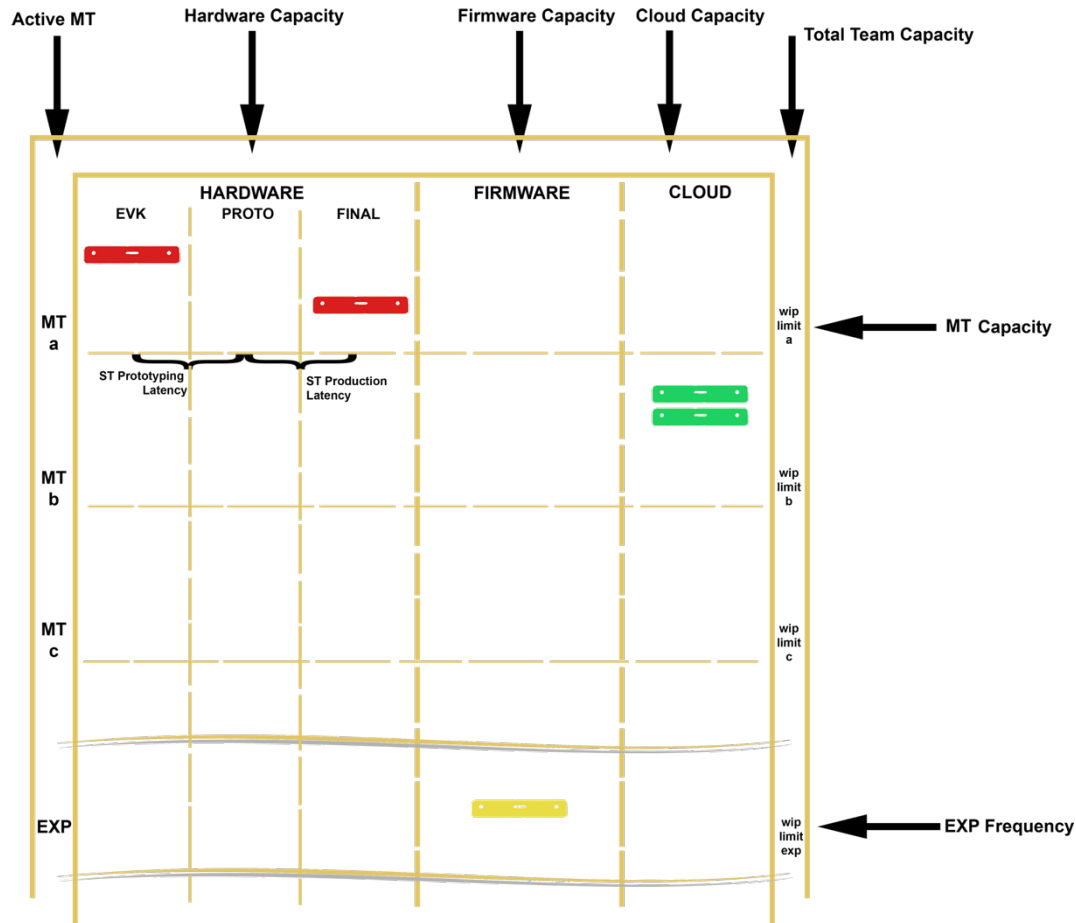


Figura 11 - D-ARCH Metrics

Entrando nel dettaglio, le metriche sono classificate in funzione della specifica dimensione afferente:

- *Dimensione Orizzontale*, relativa ai singoli Makers Team:
 - **MT Capacity**: capacità di realizzare obiettivi dello specifico MT;
 - **EXP Frequency**: frequenza di sperimentazione degli MT/ET team.
- *Dimensione Verticale*, relativa al Fiotto Team nel suo complesso:
 - **Total Team Capacity**: capacità del Fiotto Team di realizzare gli obiettivi;
 - **Hardware, Firmware e Cloud Capacity**: capacità totale del Fiotto Team di affrontare gli obiettivi annessi ai differenti aspetti IoT. La somma delle tre metriche fornisce la **Team Skills Balanced View**, ovvero la misura del bilanciamento degli skill complessivi del Fiotto Team;
 - **Active MT**: numero di MT attivi sul progetto;

- **ST (Smart Thing) Prototyping Latency:** latenza dal passaggio dall'utilizzo degli EVK al prototipo custom di Smart Thing;
- **ST (Smart Thing) Production Latency:** latenza del passaggio dal prototipo di Smart Thing alla sua versione finale.

Quando si parla di *Capacità* o *Frequenza*, l'unità minima di riferimento è chiaramente il Task.

Una nota di attenzione è dovuta nei riguardi delle “metriche di latenza”, ovvero *ST Prototyping Latency* e *ST Production Latency*: entrambe, con opportuna indicazione sulla relativa card dei tempi di inizio e completamento, consentono di evidenziare il gap tra l'utilizzo degli EVK e la realizzazione dello Smart Thing personalizzato

Queste informazioni consentono di gestire al meglio la definizione della *Bill of Materials (BOM)*, e, soprattutto, la gestione della fase di produzione fisica, spesso affidata in esterno.



6 Conclusioni

Nessuna opera è perfetta, tanto più nel mondo dello sviluppo software e hardware, dove ci sono sempre spazi di miglioramento e di evoluzione.

D'altronde proprio Agile e Lean ci hanno insegnato il mantra dell'*inspect-and-adapt* spingendoci al miglioramento continuo. *AgileIoT*, *AgileIoT Duttile* e *AgileIoT Fiotto* non sono da meno e sono in costante evoluzione, grazie alla sperimentazione diretta e ai tanti feedback provenienti dai contesti più diversi.



Author

Felice Pescatore (@felicepescatore) si occupa di Agile a livello enterprise, accompagnando le aziende nella trasformazione Culturale indispensabile per renderle più reattive e pronte alle moderne dinamiche di mercato.

Co-Authors

Questo lavoro non sarebbe potuto essere quello che è senza i consigli e il paziente supporto di: **Alessandro Alpi** (@suxstellino), **Igor Antonacci** (@bitbreaker), **Matteo Emili** (@MattVSTS), **Giuseppe Famiglietti** (@giufamiglietti), **Paolo Patierno** (@ppatierno), **Gian Maria Ricci** (@alkampfer) e **Vincent Thavonekham** (@vincentthavo).

Un sentito grazie ad **Andrea Palumbo** (@ndrea_palumbo) per il supporto negli aspetti legali, privacy in primis.





AgileIoT, AgileIoT Duttile e AgileIoT Fiotto di AgileConstellation.info è distribuito con Licenza [Creative Commons Attribuzione - Non commerciale - Non opere derivate 4.0 Internazionale](https://creativecommons.org/licenses/by-nc-nd/4.0/). L'autore, i co-autori e i contributors non possono essere ritenuti responsabili in alcun modo dell'utilizzo di quanto riportato in questo documento, in quelli annessi e nei canali digitali relativi. L'utente ha la totale responsabilità del proprio operato e libera i suddetti da ogni tipologia di incombenza diretta e indiretta.

Basato sul lavoro disponibile su **AgileConstellation.info**

Permessi ulteriori rispetto alle finalità della presente licenza possono essere disponibili presso **AgileConstellation.info**

