Orientamento allo Yocto Project

Otavio Salvador, Daiane Angolini

2016-09-07T22:00:00Z



# Sommario

# A proposito di questo opuscolo

|  |  |
| --- | --- |
| **Titolo originale** | Orientamento allo Yocto Project |
| **Licenza** | Creative Commons Attribution-ShareAlike 3.0 United States License |
| **Autore** | Otavio Salvador <[otavio@ossystems.com.br](mailto:otavio@ossystems.com.br)> |
| **Autore** | Daiane Angolini <[angolini@gmail.com](mailto:angolini@gmail.com)> |
| **Progetto grafico** | Michelle Silva <[michelleosads@gmail.com](mailto:michelleosads@gmail.com)> |
| **Versione dell'opuscolo** | Git! |

# Prefazione

Questo opuscolo è stato sviluppato in modo collaborativo. Lo sviluppo avviene su [GitHub](https://github.com/CollaborativeWritersHub/heading-for-the-yocto-project) e il contenuto è concesso in licenza in base alla [Creative Commons Attribution-ShareAlike 3.0 United States License](http://creativecommons.org/licenses/by-sa/3.0/us/).

Poiché il contenuto dell'opuscolo viene costantemente migliorato, non c'è un'errata, saremmo invece grati se si potesse segnalare qualsiasi [problema](https://github.com/CollaborativeWritersHub/heading-for-the-yocto-project/issues) trovato in modo da poterlo risolvere il prima possibile. Per contribuire, sono benvenute anche le [richieste di pull](https://github.com/CollaborativeWritersHub/heading-for-the-yocto-project/pulls) ;-)

È importante tenere a mente che il contenuto di questo opuscolo ha lo scopo di aiutare i nuovi arrivati a comprendere meglio gli obiettivi e i potenziali utilizzi dello Yocto Project, con l'intenzione di fornirne una panoramica, prima di immergersi nei dettagli tecnici su come si possono fare le cose.

# Convenzioni Utilizzate in Questo Libro

In questo libro vengono utilizzate le seguenti convenzioni tipografiche:

*Corsivo*

Indica i nuovi termini, URL, indirizzi e-mail, nomi ed estensioni di file.

Spaziatura fissa

Utilizzato per listati di programmi, nonché all'interno dei paragrafi per fare riferimento a elementi di un programma come nomi di variabili o di funzioni, database, tipi di dati, variabili di ambiente, istruzioni e parole chiave.

Spaziatura fissa grassetto

Mostra i comandi o altro testo che dovrebbe essere digitato letteralmente dall'utente.

*Spaziatura fissa corsivo*

Mostra il testo da sostituire

Questo elemento indica un consiglio o un suggerimento.

Questo elemento indica una nota generale.

Questo elemento indica un avviso o un avvertimento.

# Riconoscimenti

Le persone che hanno contribuito a questo libro sono:

TODO

generare l'elenco dei contributori

# 1. Lo Yocto Project

Sempre più fornitori di software e produttori di silicio utilizzano lo Yocto Project, portando sempre più utenti al progetto, con background diversi. Questo libro mira ad aiutare tutti coloro che sono interessati a comprendere le ragioni che hanno guidato l'adozione dello Yocto Project senza approfondire i dettagli tecnici. Gli argomenti trattati qui aiutano a:

* Evidenziare i possibili vantaggi che l'adozione dello Yocto Project può apportare ai prodotti;
* Scegliere la migliore organizzazione del team e flusso di lavoro per lo sviluppo con lo Yocto Project;
* Migliorare la comunicazione all'interno della comunità dello Yocto Project.

In definitiva, questo libro intende facilitare l'identificazione di progetti e delle aziende in cui lo Yocto Project risulta essere il framework ideale da utilizzare.

Questo primo capitolo si concentra sulla descrizione di cosa sia lo Yocto Project, le sue componenti principali e presenta la relazione con altri progetti e aziende. Tratta i seguenti argomenti:

* Cos'è lo Yocto Project;
* Quali sono le principali componenti dello Yocto Project;
* Il build Poky di sistema e le sue componenti;
* Una piccola parte della storia del progetto OpenEmbedded e di come alla fine è stato coinvolto nello Yocto Project.

## 1.1. Cos'è lo Yocto Project

Lo Yocto Project è iniziato come una collaborazione tra molti produttori di hardware, di sistemi operativi open source, di fornitori e di società di elettronica, nel 2010. La forza trainante per la creazione del gruppo di lavoro è stata la necessità di ridurre la duplicazione del loro lavoro, fornendo risorse e informazioni per utenti nuovi ed esperti. Lo Yocto Project è un gruppo di lavoro della Linux Foundation ed è definito come:

Lo Yocto Project fornisce infrastruttura e strumenti open source di alta qualità per aiutare gli sviluppatori a creare le proprie distribuzioni Linux personalizzate per qualsiasi architettura hardware, in più segmenti di mercato. Lo Yocto Project ha lo scopo di fornire un utile punto di partenza per gli sviluppatori.

~ La Linux Foundation

Il gruppo di lavoro è composto da un insieme di membri che supportano il progetto e fanno parte dell'Advisory Board. Tra questi ci sono produttori di chip, fornitori di software e progetti open source. L'hosting e la gestione dello Yocto Project da parte della Linux Foundation è, quindi, strategico per garantire l'indipendenza del progetto.

TODO

L'elenco dei membri dello Yocto Project Advisory Board (yp-ab) list: <http://bit.ly/1wXdsYI>; aggiungere la citazione.

Il gruppo di aziende che sostiene il progetto sta crescendo rapidamente, come si può vedere nella figura XXX.

TODO

Prelevare il numero dal 2010 ad oggi sui membri dell'Advisory Board e creare un grafico

In sostanza, le aziende che agiscono sullo stesso mercato si uniscono e collaborano in modo simbiotico, contribuendo a rendere il progetto compatibile con la maggior parte dei processori disponibili.

Oltre al gruppo di lavoro, lo Yocto Project è un progetto open source che fornisce template, strumenti e metodi per aiutare le aziende e gli individui a creare sistemi operativi personalizzati basati su Linux indipendentemente dall'architettura hardware. La parte open source del progetto è gestita da un collega della Linux Foundation per garantire che le decisioni tecniche non vengano prese a beneficio di un membro specifico del gruppo di lavoro.

## 1.2. Lo Yocto Project sembra una Computing Machine [Macchina Informatica]

Un modo semplice per capire quali sono i risultati dello Yocto Project è rappresentarlo come una computing machine. In breve, dobbiamo descrivere i nostri requisiti ai tool dello Yocto Project, questi vengono interpretati e viene generato l'insieme desiderato di output. Una semplice panoramica del processo svolto può essere vista in [Lo Yocto Project visto come una Macchina Informatica](#yp.xhtml#yp-input-output):

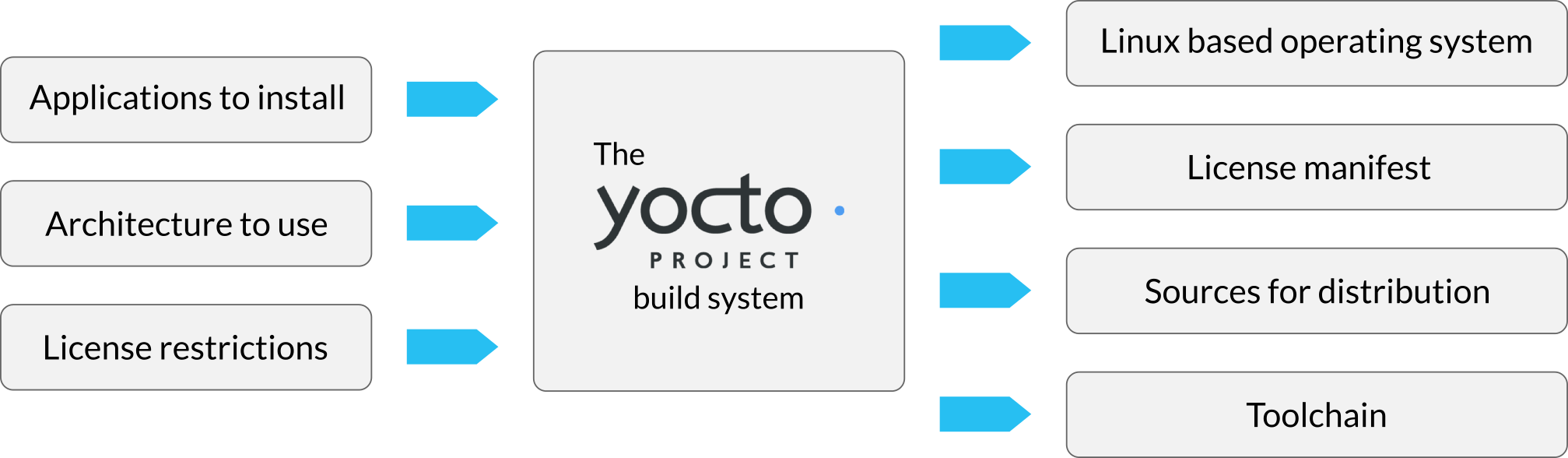


Figura 1. Lo Yocto Project sembra una Computing Machine [Macchina Informatica]

L'input è l'insieme di dati che descrivono i requisiti (ad es. elenco di applicazioni da installare, quale architettura utilizzare ecc.). Dopo che l'input è stato interpretato dagli strumenti dello Yocto Project, otteniamo uno o più output. Quando abbiamo bisogno di una scheda che esegua un sistema operativo su misura basato su Linux, possiamo ottenere tutti gli output richiesti generati direttamente per noi, tra cui:

Bootloader

Il bootloader è responsabile di inizializzare l'hardware (se necessario), caricare il kernel Linux e avviarlo. Ci sono diversi bootloader disponibili per lo Yocto Project come U-Boot, GRUB, Syslinux ecc.;

Immagine del kernel Linux

È l'immagine del kernel Linux che viene eseguita sulla macchina target [di destinazione] ed è personalizzata in base al caso d'uso previsto (ad esempio, aggiunta o rimozione di funzionalità e/o driver se richiesto);

Filesystem di root

Il filesystem di root, solitamente chiamato *rootfs*, è il bundle [pacchetto] che contiene i file, i componenti principali e le librerie necessarie per consentire l'esecuzione del sistema operativo. Inoltre, può includere altre applicazioni e file per supportare un caso d'uso specifico.

Oltre a questi, altri esempi di possibili output sono:

* elenco delle licenze dei pacchetti inclusi nel rootfs;
* I sorgenti per le distribuzioni per soddisfare i requisiti del copyleft;
* una toolchain da utilizzare con la piattaforma.

## 1.3. Il sistema di build Poky

La [computing machine] dello Yocto Project rappresentata ne [Lo Yocto Project sembra una Computing Machine [Macchina Informatica]](#yp.xhtml#yp-input-output) si riferisce al sistema di build di riferimento dello Yocto Project, chiamato *Poky*, che è composto da un insieme di file per far fare il lavoro allo strumento di orchestrazione della build.

I *sistemi di build* sono stati originariamente progettati per automatizzare le attività di build, per poi estendersi fino a includere altre attività non direttamente correlate alla compilazione, come la generazione di manifest di licenza, test a runtime automatizzato, supportati nel sistema di compilazione Poky.

Il sistema di build Poky è progettato per essere indipendente dalla piattaforma e fare un uso massiccio di ambienti di cross-compilazione. La motivazione di queste caratteristiche è quella di garantire flessibilità per lo sviluppo e l'utilizzo di diverse piattaforme target (indipendentemente dall'architettura). Il sistema di build Poky è composto dal *tool BitBake* e un insieme di default di *metadati* (cfr. [Una panoramica del sistema di build Poky](#yp.xhtml#poky-compact-overview)).

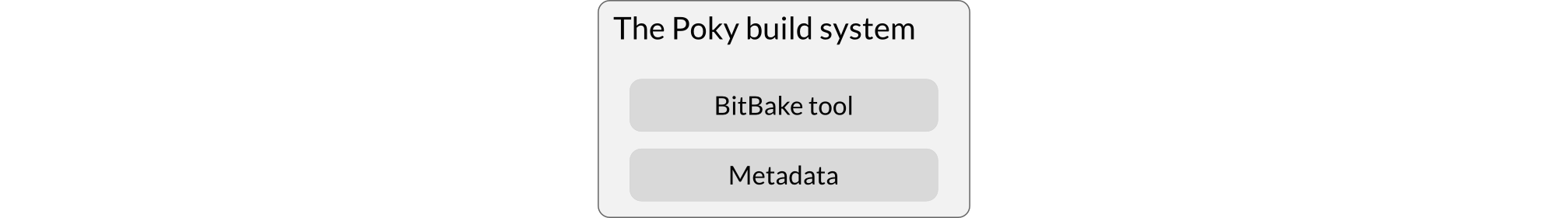


Figura 2. Una panoramica del sistema di build Poky

I metadati

È una collezione di file scritti in un mix di *Shell Script*, *Python* e un linguaggio di configurazione personalizzato, che indicano i passaggi necessari il build, il download del codice sorgente e altre attività relative a una specifica applicazione o libreria software;

Il tool BitBake

È il tool di orchestrazione della build. È responsabile di generare, ordinare ed eseguire i task [attività] in base alle informazioni nei metadati.

Questi blocchi costituiscono i meccanismi per mettere insieme migliaia di progetti open source, estensioni specifiche del fornitore o anche pezzi proprietari per formare uno stack software Linux completamente personalizzabile, completo e coerente.

### 1.3.1. Uno sguardo più da vicino al blocco dei metadati

Uno sguardo più da vicino al blocco dei metadati di [Una panoramica del sistema di build Poky](#yp.xhtml#poky-compact-overview) rivela due sotto-moduli che sono fondamentali per comprendere meglio come è composto il sistema di build Poky. I due sotto-moduli, i metadati specifici di OpenEmbedded-Core e quelli specifici dello Yocto Project, vengono mostrati ne [Il sistema di build Poky](#yp.xhtml#poky-overview):

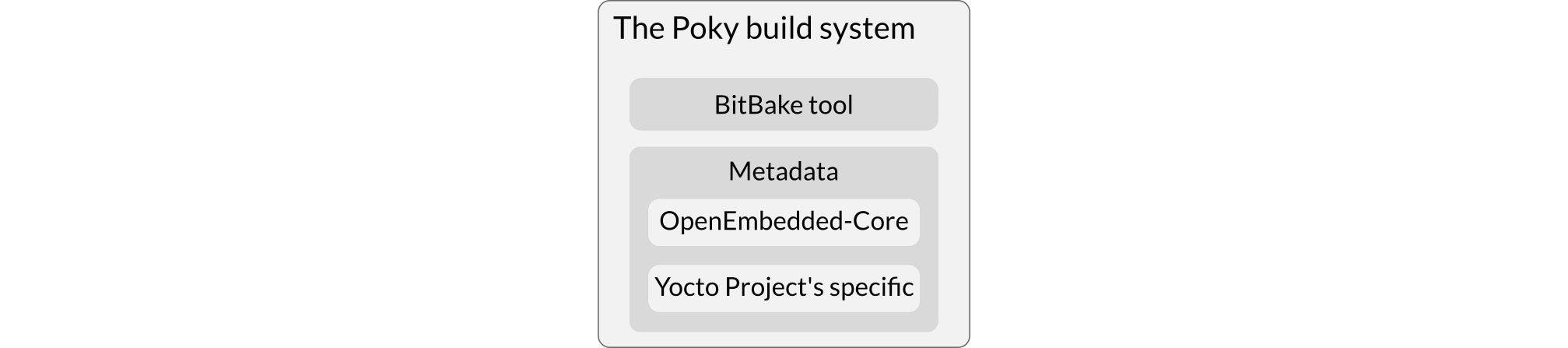


Figura 3. Il sistema di build Poky

Tali sotto-moduli possono essere intesi come:

OpenEmbedded-Core

È l'infrastruttura principale per l'ambiente di cross-compilazione e rappresenta l'insieme di base di applicazioni, librerie e utilità pronte per l'uso nei sistemi operativi basati su Linux.

Sei diverse architetture di processore (*ARM*, *ARM64*, *x86*, *x86-64*, *PowerPC*, *MIPS* e *MIPS64*) sono supportati nel sistema al momento della stesura di questo documento e tutti i test e lo sviluppo vengono eseguiti utilizzando macchine emulate, su *QEMU*.

Questo insieme di metadati è comune a tutti i diversi sistemi derivati da OpenEmbedded e uno dei più importanti è lo il Yocto Project che include OpenEmbedded-Core nel sistema di build Poky. La manutenzione è condivisa tra lo Yocto Project e il Progetto OpenEmbedded.

Specifico dello Yocto Project

Viene fornito dallo Yocto Project per configurare il sistema di build per soddisfare le sue esigenze e include una serie di *board support packages* [pacchetti di supporto della scheda] (BSP) che è il BSP di riferimento utilizzato dal Yocto Project Quality Assurance Team prima di ogni rilascio.

Lo Yocto Project, e di conseguenza il sistema di build Poky, ha una nuova versione ogni aprile e ottobre, un ciclo di rilascio di sei mesi. Questa cadenza e prevedibilità sono importanti per consentire ai fornitori di chip e di software di pianificare di conseguenza il loro programmi di rilascio e sviluppo.

## 1.4. L'alleanza tra OpenEmbedded Project e lo Yocto Project

All'incirca nel gennaio 2003, alcuni sviluppatori principali del progetto OpenZaurus hanno iniziato ad estendere la gamma di dispositivi supportati dal sistema di build, aggiungendo il supporto per altri dispositivi. Questo nuovo sistema di build è stato infine rinominato *Sistema di build OpenEmbedded*. Il sistema di build OpenEmbedded era composto da uno scheduler dei task [attività], chiamato BitBake, e da un insieme di metadati, tra cui un'enorme raccolta di software e pacchetti BSP.

BitBake, è stato ispirato e basato sul sistema di pacchetti Gentoo Portage. Lo sviluppo è stato molto agile e il progetto è cresciuto molto velocemente. Come conseguenza dello sviluppo non coordinato, era difficile utilizzare il sistema di compilazione OpenEmbedded in prodotti che richiedevano una base di codice più stabile e ordinata. Questa è stata la più grande motivazione per la creazione del sistema di build Poky.

Il sistema di build Poky è iniziato come un sottoinsieme di OpenEmbedded incentrato su una base di codice più ordinata e stabile, ma con un set limitato di BSP supportati. Le sue dimensioni ridotte hanno consentito al sistema di build Poky di iniziare a sviluppare tecnologie interessanti, come i plug-in IDE e l'integrazione con QEMU, che sono ancora in uso oggi.

Il sistema di build Poky è diventato un progetto molto interessante per i fornitori di software in quanto offriva metadati molto estensibili e uno scheduler affidabile. Questo ambiente ha consentito ai fornitori di riutilizzare e personalizzare i metadati.

In questo scenario l'indipendenza del fornitore assumeva un aspetto importante e, quindi, era urgente la necessità di un'organizzazione neutrale per gestire il tutto. Intorno a novembre 2010, è stato annunciato lo Yocto Project dalla Linux Foundation.

Lo Yocto Project è il progetto ombrello che fornisce l'infrastruttura e gestisce le risorse fornite dai suoi membri. L'indipendenza dal fornitore consente ai fornitori concorrenti di lavorare insieme in un ambiente di sviluppo comune e amichevole. Lo Yocto Project e l'OpenEmbedded Project hanno consolidato i loro sforzi su un sistema di build di base chiamato OpenEmbedded-Core, solitamente indicato come *OE-Core*.

OpenEmbedded-Core è stato creato utilizzando il meglio dei sistemi di build originali Poky e OpenEmbedded, enfatizzando e aumentando il riutilizzo di componenti, metadati e sotto-moduli formando un sistema di build *core* comune a tutti i diversi derivati da OpenEmbedded.

Anche il sistema di build Poky è stato rielaborato per utilizzare OpenEmbedded-Core. I BSP e le configurazioni extra sono stati spostati in una serie di sotto-moduli di metadati specifici dello Yocto Project. Questa rielaborazione ha facilitato una collaborazione molto più ampia tra OpenEmbedded Project e Yocto Project, riducendo la duplicazione del lavoro. Questo è il sistema di build Poky utilizzato dallo Yocto Project come sistema di build di riferimento.

L'OpenEmbedded Project e lo Yocto Project condividono la manutenzione di OpenEmbedded-Core, spingendo per un sistema di build di base molto guidato dalla comunità consentendo al contempo l'interazione attiva delle aziende attraverso i contributori dello Yocto Project e i membri dell'Advisory Board.

## Sommario

Questo primo capitolo chiarisce cos'è lo Yocto Project, qual è la differenza tra i sistemi di build Poky e OpenEmbedded-Core e come tutti questi progetti si relazionano tra loro.

Nel capitolo [[I vantaggi per i prodotti]](#X38b974c9f8b1ad1e70fb23aeee0c1450f4d415f), discutiamo delle motivazioni che possono condurre verso lo Yocto Project, indipendentemente dal fatto che si è un utente, uno sviluppatore o un imprenditore di un'azienda, trattando argomenti come i vantaggi, la riduzione del rischio e il riutilizzo della piattaforma.

# 2. I vantaggi per i prodotti

Il progetto Yocto è stato adottato da diverse aziende, organizzazioni e singoli individui. Ma perché?

In questo capitolo vengono mostrati alcuni dei motivi per cui le persone stanno adottando lo Yocto Project. Più che accelerare il time-to-market, lo Yocto Project riduce alcuni dei rischi inerenti allo sviluppo di nuovi prodotti. Alcuni degli argomenti trattati sono:

* Come lo Yocto Project permette di focalizzarsi sul prodotto;
* Perché i rischi si riducono quando si utilizza lo Yocto Project;
* Come ridurre lo sviluppo per i nuovi prodotti utilizzando lo Yocto Project;
* Quali sono i meccanismi disponibili nello Yocto Project per assistere gli uffici legali delle aziende.

## 2.1. Massimizzare il riutilizzo del lavoro

Una delle abitudini più comuni degli ingegneri è quella di reinventare la ruota. Anche se può essere un esercizio entusiasmante e uno strumento di apprendimento, a volte porta solo alla duplicazione del lavoro e allo spreco di energie. La malattia del reinvenzione della ruota è particolarmente problematica quando si verifica all'interno delle aziende, specie se non ha alcuna relazione con il suo core business.

In un passato non molto lontano i prodotti che utilizzavano Linux, come sistema operativo, avevano pochi componenti oltre al kernel. I sistemi di build utilizzati a quel tempo erano semplici e molto specializzati. Non erano progettati per diventare una piattaforma o di essere riutilizzati da più prodotti.

Con questo scenario in mente, si pensi a diverse aziende che creano i propri sistemi di build (vedi [Il periodo fertile in cui ciascuna azienda ha creato il proprio sistema di build](#X5499b7eb70371133a7af003d5fe831d9c7e87fb)), ognuna con requisiti e specifiche diverse. Questo è stato un periodo fertile, in cui sono state sviluppate diverse tecnologie e concetti importanti, a costi elevati: la duplicazione del lavoro.

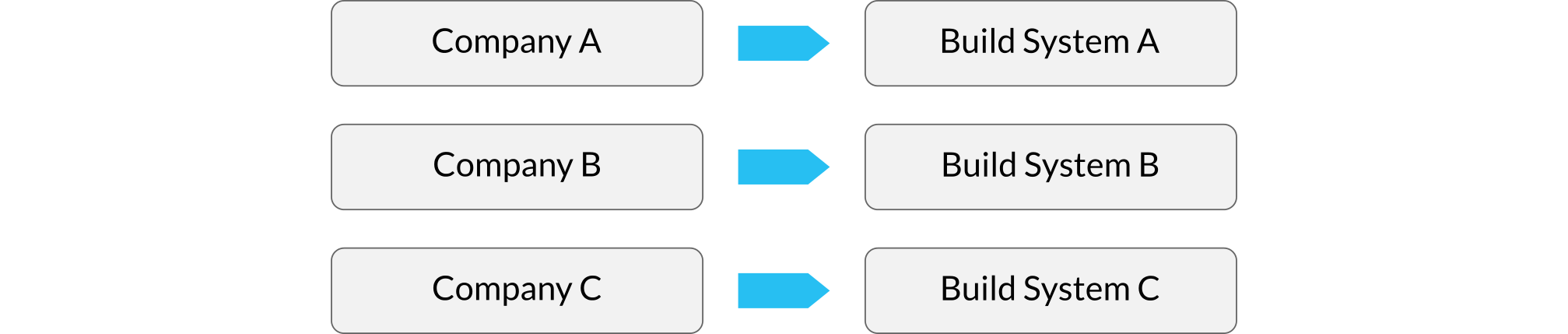


Figura 4. Il periodo fertile in cui ogni azienda creava il proprio sistema di build

Al giorno d'oggi, i sistemi operativi personalizzati non hanno praticamente limiti in termini di dimensioni e complessità, imponendo nuove sfide da superare per i sistemi di build e aumentando esponenzialmente i costi di sviluppo e manutenzione di quegli strumenti specializzati.

I sistemi di build sono comunemente sottovalutati per la loro complessità intrinseca e l'impatto sui costi di sviluppo del prodotto e il time-to-market. Sono software complessi e hanno un enorme impatto sulla quantità di lavoro per raggiungere gli obiettivi desiderati.

La scelta del sistema di build influisce in realtà sui costi di sviluppo e la manutenzione del prodotto finale poiché è una parte essenziale del nostro prodotto. Anche una semplice applicazione ha un set di istruzioni per il proprio build che può essere inteso come un sistema di build rudimentale. Più complessi sono i prodotti, più complessi sono i sistemi di build necessari (cfr. [Ogni prodotto ha un sistema di build](#Xd9eb8992b218c7023bd137a2e14ec4518a24b05)).

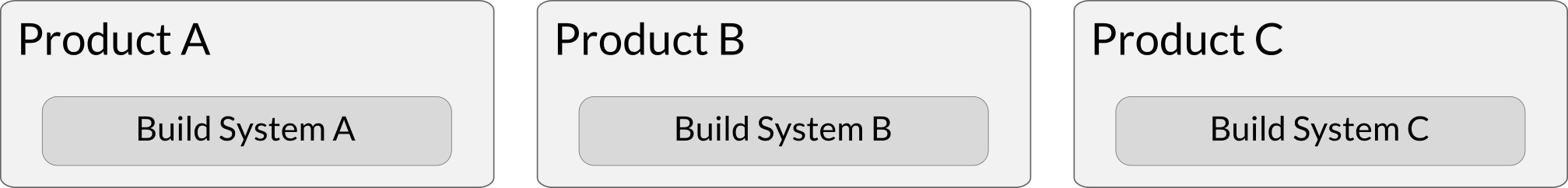


Figura 5. Ogni prodotto ha un sistema di build

A causa della complessità sempre crescente dei sistemi operativi, sono stati sviluppati diversi sistemi di build - alcuni open source, altri proprietari - concentrandosi su diversi sottoinsiemi di requisiti. Lo Yocto Project non è stato il primo tentativo di risolvere la complessità, la duplicazione del lavoro e il problema dei costi, ma uno dei più riusciti finora.

### 2.1.1. Non concentrarsi sul sistema di build

Per un'azienda aveva senso non preoccuparsi del sistema di build, in passato, quando il sistema target non era così complesso. Quando i prodotti avevano una dipendenza minima dal software, i sistemi di build giocavano un ruolo secondario nello sviluppo dei prodotti.

Se il core business della propria azienda non ha alcuna relazione con lo sviluppo del sistema di build, non ci si dovrebbe preoccupare del sistema di build, ma concentrarsi sul prodotto.

Di solito, i prodotti sono progettati per risolvere una serie di requisiti, provenienti sia da un'intuizione che dal team del marketing, e coinvolgono diverse applicazioni. Il sistema di build gioca un ruolo essenziale nel soddisfare tutti questi requisiti. Alcuni sono soddisfatti da applicazioni specifiche mentre altri sono correlati al sistema di build stesso (cfr. [L'impatto del sistema di build sul prodotto](#Xd761b55ca65d17a897c995c3e049377d0054cd0)).

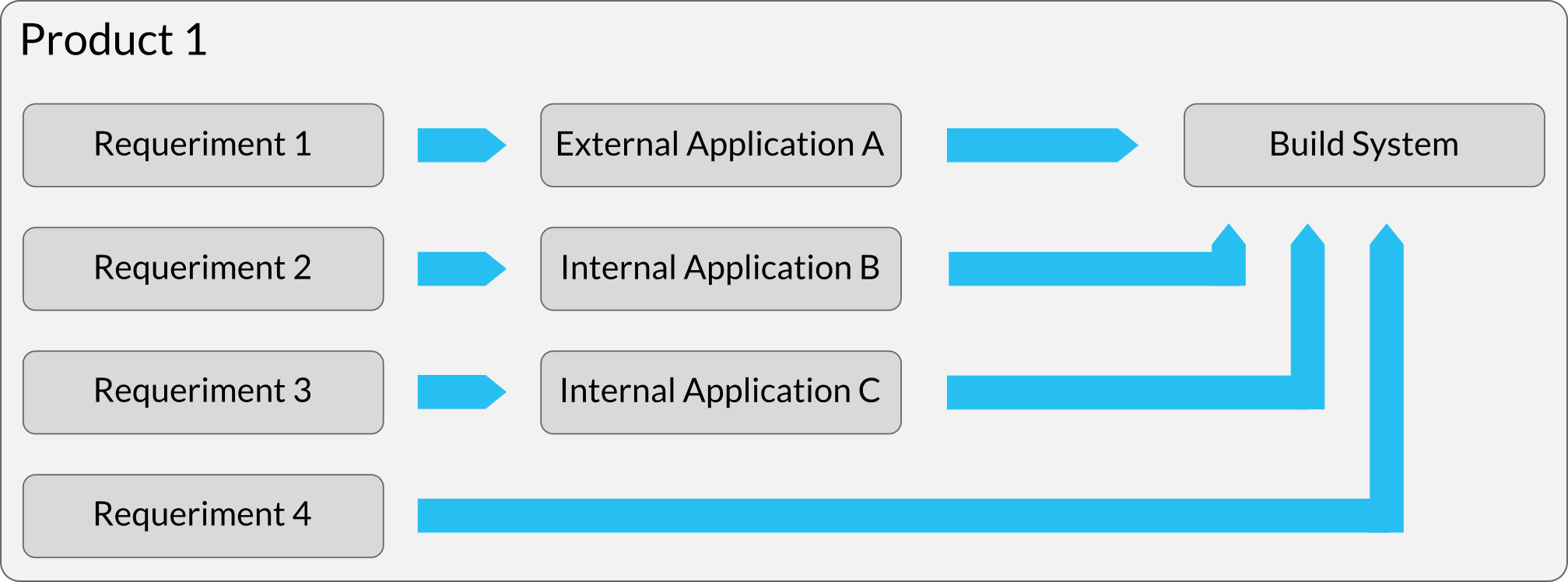


Figura 6. L'impatto del sistema di build sul prodotto

Il sistema di build è uno tool necessario che dovrebbe servire per i prodotti. Dovrebbe aiutare il team e l'azienda ad evitare le solite insidie, indirizzare le funzionalità principali direttamente al punto e non perdere tempo a gestire lo stesso problema più e più volte.

### 2.1.2. Serve un ampia base d'uso

Con l'obiettivo di ridurre la duplicazione del lavoro, molte aziende e sviluppatori hanno iniziato a concentrarsi sui loro sistemi di build preferiti, migliorandoli secondo le necessità. Lo Yocto Project, essendo uno di questi, ha iniziato a essere visto come un'opzione sufficientemente flessibile per aggregare diversi presupposti e obiettivi. Ciò ha portato a un'ampia adozione tra gli sviluppatori (cfr. [Diverse aziende hanno adottato il progetto Yocto](#X6b3f45b789deedef18e6f8c3abf88df091fe5d4)).

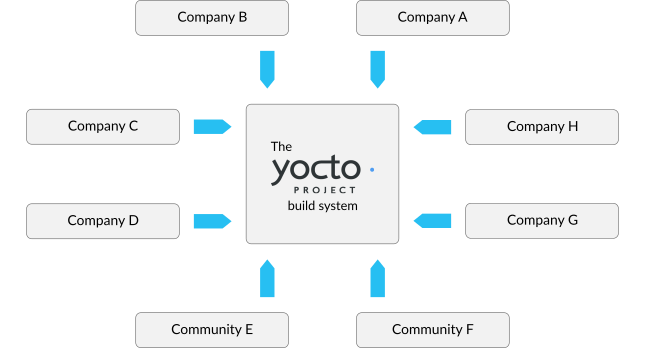


Figura 7. Diverse aziende hanno adottato il progetto Yocto

Dal punto di vista del business, l'ampia adozione aiuta a trovare aziende esperte, ad offrire servizi e consulenza, sviluppatori e utenti sul campo. Questo è un fattore chiave nella scelta di un nuovo sistema di build.

Un'importante conseguenza di una massiccia adozione dello Yocto Project è l'ampia serie di requisiti diversi che accetta la piattaforma per garantirne la flessibilità e garantire una solida base di lavoro per tantissimi prodotti.

La gigantesca comunità formata attorno allo Yocto Project aumenta la sua velocità di sviluppo, l'insieme di funzionalità supportate e il numero di fornitori di chip, di fornitori di software e progetti di comunità che sviluppano prodotti sulla stessa piattaforma.

## 2.2. Riduzione del rischio

Ci si assumono sempre dei rischi. Come nella vita, si dovrebbero limitare lavorando allo sviluppo del prodotto. Lo Yocto Project contribuisce a ridurre dei rischi.

Il *riutilizzo della piattaforma del sistema di build*, come conseguenza dell'utilizzo dello Yocto Project, consente al prodotto di fare affidamento su una piattaforma di sviluppo solida e testata utilizzata da diverse aziende e sviluppatori. Come diceva Isaac Newton, possiamo vedere più lontano stando sulle spalle dei giganti.

Oltre al riutilizzo della piattaforma del sistema di build, lo Yocto Project consente di progettare una piattaforma per il prodotto. La *piattaforma del prodotto* consente di costruire le fondamenta per lo sviluppo del prodotto riducendo lo sviluppo di ogni nuovo prodotto (cfr. [Lo Yocto Project utilizzato come piattaforma del prodotto](#X42b1d8f1ec8b4c37f58c42fc48e79d09f66fdb0)), facilita la conformità agli standard aziendali e alle ipotesi generali sui prodotti, come la grafica, le librerie di base aziendali e così via.

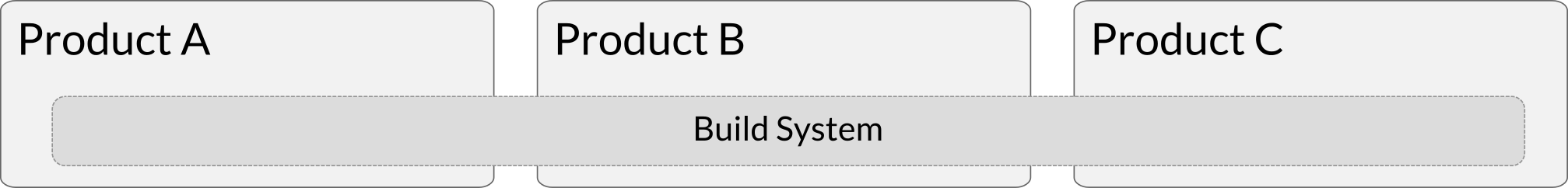


Figura 8. Lo Yocto Project utilizzato come piattaforma del prodotto

Confrontando [Ogni prodotto include un sistema di build](#Xd9eb8992b218c7023bd137a2e14ec4518a24b05) e [Lo Yocto Project utilizzato come piattaforma di prodotto](#X42b1d8f1ec8b4c37f58c42fc48e79d09f66fdb0), le differenze su come il sistema di build si adatta al prodotto sono chiare. Quando condividiamo il sistema di build tra due o più prodotti, si riducono significativamente i costi e i tempi dello sviluppo nel sistema di build.

La flessibilità dello Yocto Project è essenziale quando lo stesso sistema di build viene condiviso tra più prodotti. Fornisce gli strumenti necessari per soddisfare i requisiti specifici del prodotto, facilitando il riutilizzo di quelli in comune.

Il sistema di build dello di Yocto Project utilizzato come piattaforma del prodotto insieme alla sua flessibilità costituiscono la combinazione perfetta per la riduzione dei rischi.

Una delle caratteristiche progettuali più importanti dello Yocto Project è la riproducibilità. Gli strumenti dello Yocto Project producono lo stesso output ad ogni esecuzione e forniscono diversi meccanismi per verificarlo. La riproducibilità consente di realizzare in modo affidabile prodotti basati sugli strumenti dello Yocto Project e di scalare lo sviluppo tra i diversi team poiché il risultato della build è sempre lo stesso.

L'ampia adozione che si vede nella 'community' dello Yocto Project è un altro importante punto di riduzione del rischio. Questa community aiuta a mantenere molto solide le versioni stabili dello Yocto Project, incluse correzioni di bug e gli aggiornamenti sulla sicurezza quando ci sono nel mentre sviluppa attivamente la versione successiva, in modo che la piattaforma sia all'altezza delle sfide di domani.

## 2.3. Conformità e gestione delle licenze

Lo Yocto Project contribuisce all'aderenza alla conformità e alla gestione delle licenze nei prodotti. Non indica su cosa è necessario per essere conformi a una o più licenze, ma produce una serie di risultati che aiutano il team legale e l'ufficio tecnico a farlo.

In sostanza, aiuta a:

* Intercettare le inosservanze alle licenze
* Elencare le licenze in un prodotto
* Raccogliere il codice sorgente per la distribuzione, se la licenza lo richiede
* Evitare l'uso di una licenza specifica in un prodotto (es. GPLv3)

Lo Yocto Project ha alzato il livello nello scenario degli strumenti di supporto alla conformità delle licenze. Gli output che genera sono molto utili quando c'è da verificare la conformità di tutte le licenze incluse nel progetto. I tool dello Yocto Project non sostituiscono l'ufficio legale ma ne facilitano il lavoro.

## Sommario

L'obiettivo principale di questo capitolo è quello di descrivere perché utilizzare lo Yocto Project e spiegare perché ci sono così tanti sviluppatori e aziende che lo adottano.

Vengono brevemente citati gli ausili per la conformità e la gestione delle licenze disponibili nello Yocto Project, per mostrare come possono essere d'aiuto alle aziende sotto diversi aspetti, non solo quelli tecnici. Altri argomenti vengono discussi in modo più dettagliato, come il modo in cui lo Yocto Project consente di concentrarsi sul prodotto, a ridurre i rischi e ridurre il costo dello sviluppo.

Il prossimo capitolo [[Orientamento allo Yocto Project]](#X69ff04039a63a1d701edeef77d218ddd999c518) descrive alcuni flussi di lavoro per utilizzare lo Yocto Project. Questo è particolarmente importante in quanto ci indirizza nella scelta della base giusta quando si adotta lo Yocto Project per il progetto o per l'azienda. Include anche alcune delle 'best practices' adottabili durante questo processo.

# 3. Orientamento allo Yocto Project

A prima vista, il modo in cui le aziende e i progetti possono utilizzare lo Yocto Project non è ovvio perché è molto flessibile e si adatta a team di diverse dimensioni e a diversi flussi di lavoro, e solitamente è visto solo come uno strumento di automazione delle attività.

In questo capitolo analizzeremo alcuni possibili flussi di lavoro e forniremo dei suggerimenti per l'organizzazione di team e progetti di diverse dimensioni, in modo da adattare la mentalità comune di sviluppo software al framework dello Yocto Project. In questo capitolo vengono trattati i seguenti argomenti:

* Il comune flusso di lavoro dello sviluppo software
* Un approccio per organizzare lo sviluppo del software
* Una visione pratica sull'applicabilità dei ruoli nel framework dello Yocto Project
* I possibili vantaggi di un uso strategico dell'outsourcing

## 3.1. La pianificazione per lo Yocto Project

Esistono diversi approcci nello sviluppo del software. Una fase comune, nella maggior parte di essi, è la specifica delle funzionalità, che è fondamentale per la comprensione di tutti i requisiti coinvolti.

Uno dei maggiori vantaggi dello Yocto Project è la flessibilità. Questo si nota non solo sul sistema di build stesso, ma anche su come si può organizzare l'uso dello Yocto Project nel team.

Tutti i possibili requisiti, si possono disporre in tre gruppi:

Requisiti del marketing

I requisiti di prodotto che provengono dalle unità aziendali, dai team gestionali o di marketing, qui possono adattarsi facilmente. Ad esempio, le caratteristiche generali del prodotto relative alla segmentazione del mercato, i vincoli legali e così via.

Requisiti dell'applicazione

L'insieme dei requisiti necessari per ottenere il funzionamento e l'aspetto del prodotto che dipendono dalla personalizzazione dell'applicazione o dal nuovo sviluppo. Questi sono responsabili di soddisfare i requisiti del marketing.

Requisiti del sistema di build

Ci sono diversi aspetti che hanno un forte impatto sullo sviluppo del prodotto che fanno parte della piattaforma di base o degli strumenti di sviluppo. Di solito sono correlati al sistema di build o a un artefatto da esso generato e possono includere l'architettura target, la configurazione della toolchain, i driver di accelerazione hardware, il set di applicazioni richieste e altro ancora.

Questi aspetti sono principalmente guidati dai requisiti sia del marketing che applicativi.

Specificare un software utilizzando una lista dei requisiti, consente di avere l'elenco delle attività per lo sviluppo del software definito in base a tali requisiti.

Il numero di sviluppatori in un team può variare notevolmente, tuttavia le attività da completare per lo sviluppo del prodotto sono quasi le stesse, indipendentemente dalle dimensioni del team. La differenza sta nell'organizzazione del team e nella separazione dei compiti. I compiti sono essenzialmente:

* Preparazione della board [scheda hardware];
* Sviluppo del bootloader;
* Sviluppo del kernel;
* Sviluppo del device driver;
* Sviluppo dell'applicazione;
* Personalizzazione dell'applicazione;
* Gestione dell'elenco di applicazioni;
* Sviluppo del *Software Development Kit* (*SDK*);
* Integrazione nel sistema di build;
* Rilasci;
* Gestione della licenza.

Elenca le attività più comuni coinvolte nella parte software di un prodotto basato su Linux. Quando si raggruppano queste attività nei ruoli dello sviluppatore, è possibile vedere una possibile disposizione ne [Le attività per lo sviluppo del software sono suddivise in quattro ruoli diversi](#Xe809bbfc1964b6226feb9d6656b2ba46f1397f3).

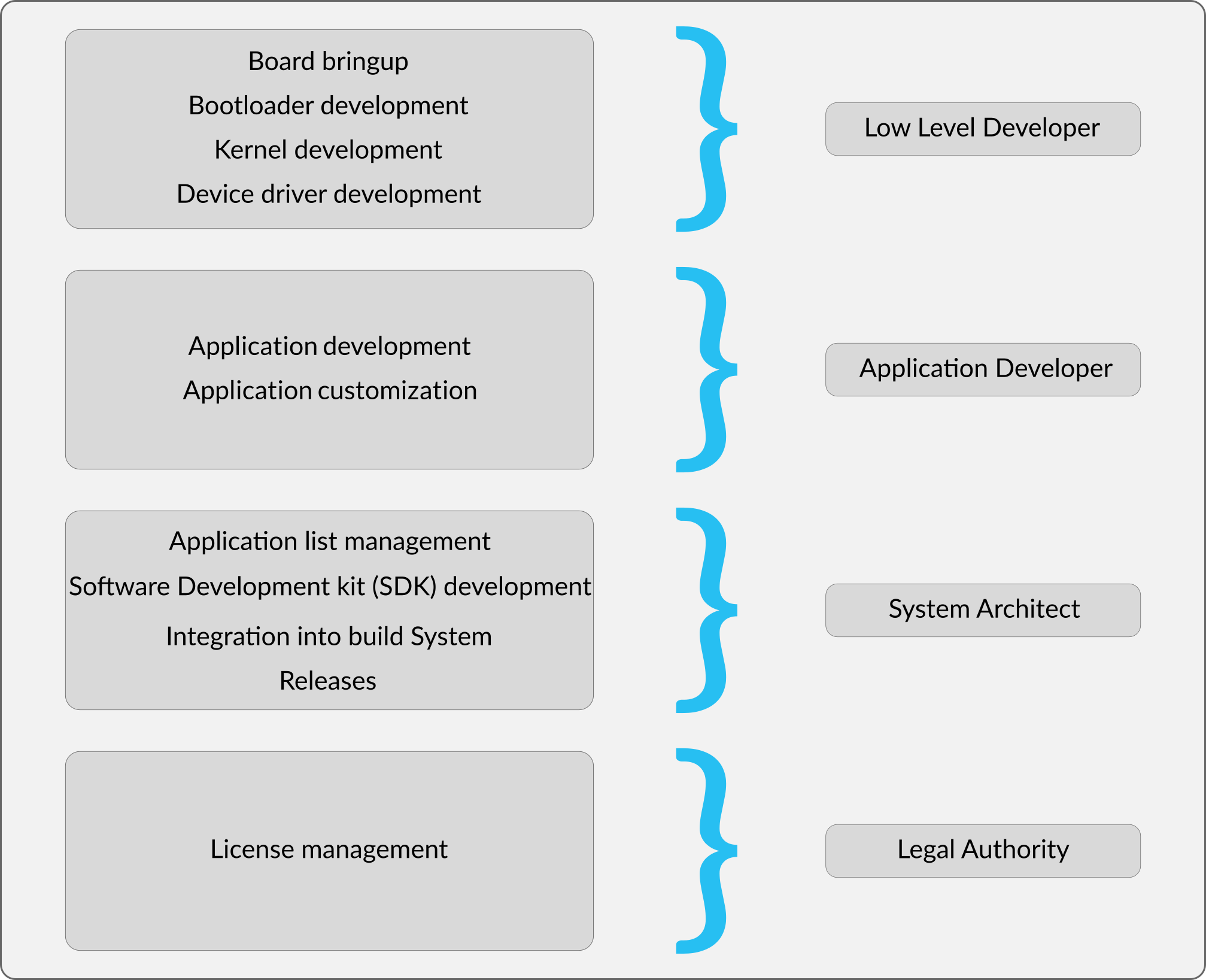


Figura 9. Le attività per lo sviluppo del software sono suddivise in quattro ruoli diversi.

Tali ruoli possono essere descritti come:

Sviluppatore Low Level

Esegue attività relative al kernel Linux e al bootloader, sviluppo o personalizzazione di driver e altre attività strettamente connesse all'hardware stesso e ai suoi componenti.

Sviluppatore di Applicazione

Si focalizza sul building e sull'adattamento delle applicazioni specifiche del prodotto. Può avere un ampio bagaglio di conoscenze applicative, poiché ogni prodotto può utilizzare una tecnologia applicativa diversa. Questo ruolo è molto importante in quanto agisce sul punto della differenziazione del prodotto.

Architetto di Sistema

Il punto di contatto centrale sia per lo sviluppatore di basso livello che per lo sviluppatore di applicazioni. È responsabile di:

* Validare i requisiti
* Fornire gli strumenti necessari per lo sviluppo
* Integrare continuamente il lavoro degli Sviluppatori Low Level e di quello degli Sviluppatori di Applicazioni
* Preparare immagini di test del lavoro integrato per il Controllo Qualità, i Test di Regressione e la Validazione
* Preparare i rilasci di produzione dei sistemi di build

Authority Legale

Quello con la responsabilità legale di determinare i vincoli legali e gli obiettivi del prodotto.

Questi ruoli richiedono un alto grado di comunicazione quando vengono eseguiti da diversi membri del team, come si può vedere in [Comunicazione tra i ruoli](#X2b3abc634f4e218169df0d3b7ca8420f0e52f72).

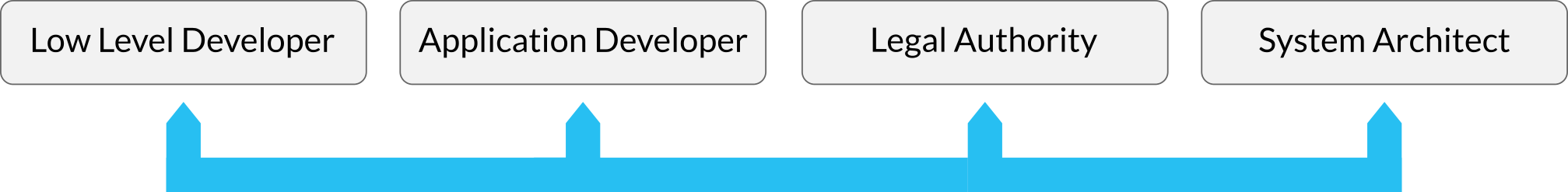


Figura 10. Comunicazione tra i ruoli

Lo stesso approccio utilizzato per lo sviluppo del software può essere utilizzato nello sviluppo del software utilizzando lo Yocto Project. Il tipo di requisiti e l'elenco delle attività non cambia di molto.

Tali ruoli si possono assegnare a uno o più membri del team e dipendono fortemente dalle dimensioni del team. Lo Yocto Project assiste l'ottimizzazione dell'organizzazione del team, sia che sia composto da due che da da cento sviluppatori.

### 3.1.1. Miglioramento del flusso di lavoro dello sviluppo del software

I ruoli appena presentati vengono eseguiti in tutto il ciclo del progetto, sebbene la loro assegnazione ai membri del team possa variare nel tempo. I tool dello Yocto Project aiutano con la possibile separazione dei ruoli, essendo abbastanza flessibili da essere utilizzati in un'ampia varietà di progetti.

I team molto piccoli generalmente concentrano tutte le attività di sviluppo su un numero limitato di sviluppatori. Di solito si hanno piccoli progetti con una sola persona che svolge tutte le attività dello sviluppo. D'altra parte, i progetti più grandi richiedono la parallelizzazione del lavoro e la separazione delle responsabilità, che sono una delle forze trainanti per ridurre il "time to market" dei nuovi prodotti.

La separazione dei ruoli può essere facilitata (o meno) dalla scelta del sistema di build. Alcuni sistemi di build presuppongono la sovrapposizione dei ruoli (ad es. Sviluppatore Low Level e Architetto di Sistema).

Nel caso dello Yocto Project, è chiaro che la strategia di separazione dei ruoli fa parte dei suoi principi progettuali ed è stata portata ad un livello elevato. Il modo in cui viene applicato a un progetto dipende da come si organizza il team. Questo dà la libertà di scegliere come organizzare il team, tenendo conto del progetto e delle caratteristiche specifiche del team. Ora la sovrapposizione dei ruoli è utilizzabile a seconda dell'opportunità, non a causa delle limitazioni degli strumenti del sistema di build.

I ruoli si possono mappare al tipo di artefatti o ai componenti che coprono, come si può vedere nella tabella:

Tabella 1. Mappa dei ruoli e delle attività

| Ruolo | Descrizione |
| --- | --- |
| **Sviluppatore Low Level** | Usa la toolchain generata dall'Architetto di Sistema |
| **Sviluppatore di Applicazione** | Usa l'SDK con la toolchain, le librerie e le utilità dell'applicazione. L'SDK viene generato e manutenuto dall'Architetto di Sistema |
| **Architetto di Sistema** | Integra il lavoro svolto dallo Sviluppatore Low Level e dallo Sviluppatore dell'Applicazione, manutiene la toolchain e l'SDK, genera il materiale per il controllo della conformità delle licenze da parte dell'autorità legale, generare gli artefatti per la conformità delle licenze, lavora nel prodotto generale e coordina i rilasci. |
| **Authority Legale** | Definisce i vincoli legali e gli artefatti necessari alla conformità delle licenze, collaborando con tutti gli altri ruoli per soddisfare i requisiti necessari e cercare alternative. |

I ruoli hanno comunemente una gerarchia intrinseca, come mostrato in [La gerarchia proposta](#X05b7b6588c73336802f053da65e317e007b03ae). Può variare da un'organizzazione all'altra, tuttavia tenerlo in considerazione aiuta nell'organizzazione del team.

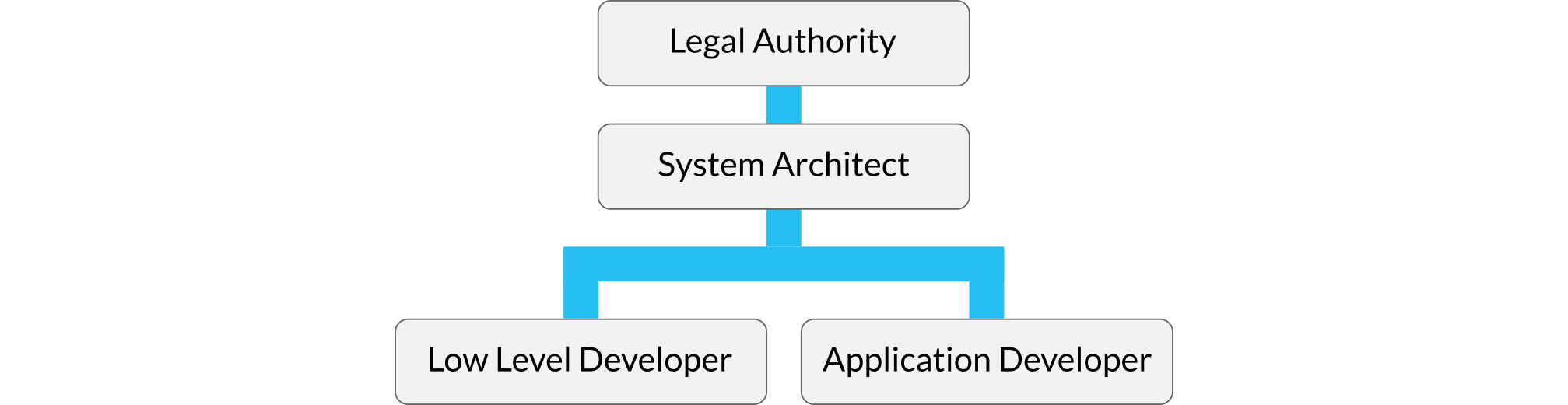


Figura 11. La gerarchia proposta

In questo scenario, si possono facilmente includere esperti di dominio. Il ruolo assegnato all'esperto di dominio dipende dalle competenze cercate.

Un esperto di streaming multimediale può fungere da Sviluppatore di Applicazioni nel caso in cui il progetto necessiti di un miglioramento delle prestazioni di streaming multimediale o lavorare come consulente dell'Authority legale quando serve un'alternativa per un vincolo legale di un codice multimediale.

Il design dello Yocto Project facilita la separazione e la sovrapposizione dei ruoli. Siamo noi a decidere come organizzare al meglio il team in base al progetto o alle esigenze aziendali.

## 3.2. Outsourcing [Esternalizzazione]

Uno dei grandi vantaggi dell'utilizzo dello Yocto Project è la possibilità di scegliere ciò che meglio si adatta al progetto.

A volte si vuol mantenere tutto lo sviluppo internamente e questa è un'opzione perfetta. Tuttavia, a seconda dei requisiti del progetto, si si potrebbe ritrovare con risorse o conoscenze limitate su un'area specifica. È qui che l'outsourcing viene in aiuto.

La strategia di outsourcing può consentirci di rispettare una scadenza difficile o alleviare il team tecnico a concentrarsi sui punti peculiari del prodotto.

Un'attività comunemente esternalizzata è lo sviluppo del driver di uno specifico dispositivo, non ancora supportato dal kernel Linux, riducendo probabilmente il costo totale.

Anche se l'esternalizzazione dei ruoli proposti può sembrare improbabile, potrebbe avere molto senso in quanto i ruoli raggruppano le attività che potrebbero non essere al centro del prodotto o del team. Il ruolo di Authority Legale è il più facile da vedere come candidato per l'outsourcing in quanto una società legale può avere un'esperienza più ampia in termini legali e il prodotto non ha il suo valore nel suo attributo legale. Però, non è così chiaro capire che forse il ruolo di Architetto di Sistema si adatterebbe anche all'outsourcing.

L'ecosistema formatosi attorno allo Yocto Project facilita l'esternalizzazione quando necessario, in quanto diverse aziende sono in grado di offrire servizi e prodotti che utilizzano gli strumenti dello Yocto Project e le sue linee guida.

L'Architetto di Sistema dovrebbe avere la conoscenza interna degli strumenti dello Yocto Project per apportare le modifiche necessarie al prodotto e fare la scelta tecnica migliore. In alcuni progetti specifici, affidarsi a un'azienda con questa competenza interna può accelerare lo sviluppo della piattaforma del prodotto e ridurre il rischio nelle decisioni relative ad essa.

## Sommario

L'organizzazione dello sviluppo del software può risultare difficile. Essendo molto flessibile e adattandosi bene a progetti e team di diverse dimensioni, lo Yocto Project contribuisce in modo positivo.

In questo capitolo si capisce come applicare il framework dello Yocto Project al flusso di lavoro dello sviluppo software, raggruppando le attività necessarie tra i membri del team in modo da utilizzarne al meglio la flessibilità e la potenza. Si esaminano i motivi per cui l'outsourcing può essere vantaggioso per un progetto e come lo Yocto Project aiuta in questo.

Il prossimo capitolo [[L'ecosistema della community dello Yocto Project]](#X8741ada44f50a3184b941227235243c6498d701) descrive l'ecosistema della comunità dello Yocto Project e presenta alcune indicazioni su come parteciparvi, in modo che i nuovi arrivati possano sentirsi più a loro agio a farne parte.

# 4. L'ecosistema della community dello Yocto Project

Uno dei maggiori vantaggi dello Yocto Project è il suo ecosistema di community. Per parteciparvi è importante sviluppare la capacità di comunicare correttamente e di contribuire in modo sinergico.

Questo capitolo raccoglie:

* I principali canali di comunicazione utilizzati nella community
* Dove segnalare o trovare informazioni su problemi noti
* L'etichetta comportamentale [galateo] prevista

## 4.1. Cos'è una community

Fornire una definizione per una community è limitante. Una community [comunità] è un organismo in movimento e la sua percezione varia a seconda del background e delle precedenti esperienze con altre comunità. Ad esempio, alcune persone credono che la community sia composta solo dalle persone che le inviano codice, altri vedono anche gli utenti e i partner come parte della comunità.

Una definizione generale per una community può essere: un insieme di persone con un obiettivo comune.

La maggior parte delle community ha molti aspetti comuni e dopo aver preso dimestichezza all'interno della prima community e averne capito gli aspetti sociali intrinseci, diventa molto più facile partecipare alle altre.

Le community possono creare molti prodotti diversi come codice sorgente, documentazione, risultati dei test, dipendenti da ciascuna community. Nel caso dello Yocto Project, la community esiste attraverso i molteplici canali di comunicazione (come mailing list, canali IRC, conferenze) e produce codice sorgente, documentazione e risultati dei test per diversi progetti.

## 4.2. I canali di comunicazione dello Yocto Project

La community dello Yocto Project ha molte somiglianze con la community del kernel Linux poiché riunisce aziende, altri progetti open source e singoli individui. Per qualcuno abituato a partecipare, può essere chiaro come agire, tuttavia è importante tenere presente che ogni community ha le sue particolarità e i suoi canali di comunicazione.

Le persone coinvolte nello sviluppo dello Yocto Project sono distribuite in tutto il mondo. Gli incontri faccia a faccia, che sono comuni nella maggior parte dei grandi progetti open source, sono il più delle volte impossibili. A volte, ci sono eventi globali e le persone si incontrano, ma la comunicazione quotidiana viene effettuata utilizzando le mailing list.

L'elenco di tutte le mailing list della community, comprese quelle relative al BSP, è su: <https://www.yoctoproject.org/tools-resources/community/mailing-lists>

Lo Yocto Project, essendo un ombrello per diversi progetti, ha diverse mailing list per ciascuno di questi progetti. Le mailing list più importanti sono visibili nella tabella [[Le mailing list dello Yocto Project]](#X32c896a5035cda83bb6f28b32743b54b17d68de).

Tabella 2. Le mailing list dello Yocto Project

|  |  |
| --- | --- |
| Nome | Descrizione |
| [Yocto Project Discussions](https://lists.yoctoproject.org/listinfo/yocto) | Mailing list di discussione generale sullo Yocto Project. Se non si sa quale mailing list usare, usare questa. |
| [Poky Discussions](https://lists.yoctoproject.org/listinfo/poky) | Discussioni sul sistema di build Poky e le patch riguardanti la distribuzione Poky. |
| [OpenEmbedded Core Developers](http://lists.openembedded.org/mailman/listinfo/openembedded-core) | La mailing list di OpenEmbedded-Core. È qui che vengono discussi lo sviluppo degli aspetti fondamentali del sistema di compilazione e vengono inviate le patch. |
| [BitBake Developers](http://lists.openembedded.org/mailman/listinfo/bitbake-devel) | La mailing list specifica di BitBake per la discussione generale sulle nuove funzionalità, problemi e patch. |
| [Toaster](https://lists.yoctoproject.org/listinfo/toaster) | Discussione e patch dell'interfaccia web per BitBake. |

Come in ogni gruppo sociale, si dovrebbero rispettare alcune linee guida per evitare imbarazzi interagendo con altri membri della community.

Nelle grandi community open source, il rispetto delle linee guida aiuta molto a integrarsi facilmente nella comunità e attirare l'attenzione degli altri. È molto comune che quando incontrando un problema si abbia l'impulso di chiedere aiuto immediatamente. Tuttavia, per mostrare rispetto verso gli altri membri della community, si dovrebbe almeno fare una ricerca per una soluzione nei normali motori di ricerca e archivi delle mailing list, per non porre la stessa domanda a cui è stata già data risposta.

Quando non si riesce a trovare una risposta ad un problema, le mailing list della community possono essere di aiuto. Qui si devono tenere a mente alcune linee guida, semplici ma importanti:

Semplicità

Spiegare in modo semplice il problema che riscontrato e una semplice descrizione passo passo su come riprodurlo. Le persone nella lista possono indicare se è necessario o meno il log — non inondare la mailing list con tanti allegati, log formattati e immagini.

Gentilezza

Le persone nella mailing list non hanno il dovere di aiutare. Quando lo fanno, lo fanno per benevolenza.

Semplificare la vita a chi legge

Invece di fare un "top-post", lo stile della risposta "in linea" rende la comunicazione molto più semplice ed efficace. È anche molto consigliato l'uso di testo semplice e del Copia e Incolla (per evitare di allegare log e immagini).

Queste linee guida sono solo un sottoinsieme di ciò che le persone più apprezzano nella community dello Yocto Project. È possibile accedere alla versione community sul [wiki dello Yocto Project](https://wiki.yoctoproject.org/wiki/Community_Guidelines).

## 4.3. Documentazione

La documentazione dello Yocto Project è perfetta come riferimento a lungo termine, quando si è consapevoli di ciò che si sta cercando e e si ha una conoscenza di base dei suoi tool. Questa serie di documenti, insieme a questo libro, sono una potente combinazione da avere con sé durante le immersioni nello Yocto Project.

Una raccolta di documentazione ben mirata è disponibile nello Yocto Project ed è visualizzabile sul [sito web](http://www.yoctoproject.org/documentation) e quelle più comunemente utilizzate sono:

[Yocto Project Quick Start](https://www.yoctoproject.org/docs/2.4.4/yocto-project-qs/yocto-project-qs.html)

È probabile che sia il primo contatto con la documentazione del progetto in quanto presenta le informazioni necessarie per iniziare rapidamente e iniziare a costruire un'immagine.

[Yocto Project Development Manual](http://www.yoctoproject.org/docs/1.8/dev-manual/dev-manual.html)

Fornisce una panoramica delle attività comuni per lo sviluppo del kernel e dello spazio utente. In realtà, raccoglie riferimenti ad altre fonti in cui si possono trovare informazioni dettagliate.

[Yocto Project Linux Kernel Development Manual](http://www.yoctoproject.org/docs/1.8/kernel-dev/kernel-dev.html)

Descrive il framework Yocto Project Linux Kernel, come usarlo e personalizzarlo.

[Yocto Project Board Support Package (BSP) Developer’s Guide](http://www.yoctoproject.org/docs/1.8/bsp-guide/bsp-guide.html)

Definisce le linee guida generali e la standardizzazione per metadati BSP, strumenti e concetti.

[Yocto Project Application Developer’s Guide](http://www.yoctoproject.org/docs/1.8/adt-manual/adt-manual.html)

Fornisce informazioni per iniziare con l'SDK dell'applicazione. Pratico per sviluppare applicazioni a livello di utente e "low-level".

[BitBake User Manual](http://www.yoctoproject.org/docs/1.8/bitbake-user-manual/bitbake-user-manual.html)

Fornisce l'utilizzo e il materiale di riferimento, inclusa la sintassi dei metadati, per il tool BitBake.

[Yocto Project Reference Manual](http://www.yoctoproject.org/docs/1.8/ref-manual/ref-manual.html)

Fornisce un riferimento allo Yocto Project, comprese variabili, classi e altre informazioni sui metadati che sono di grande valore durante l'uso quotidiano dei tool. Indubbiamente il manuale più usato.

[Yocto Project Profiling and Tracing Manual](http://www.yoctoproject.org/docs/1.8/profile-manual/profile-manual.html)

Questo documento descrive come sfruttare gli strumenti di profilazione e tracing disponibili.

Ci sono diversi vantaggi nell'entrare in una community, alcuni possono essere personali altri professionali. È importante che si trovi il modo personale per divertirsi e contribuire alla comunità che abbia un senso e sia allineato con gli obiettivi. Un consiglio, quando si entra in una nuova community, è quello di seguire la mailing list target per sapere come si comporta quella community, prima di iniziare effettivamente a contribuire e, alla fine, diventarne parte.

## Sommario

Questo capitolo può sembrare che 'sfondi una porta aperta' per sviluppatori di Software Libero più esperti o professionisti di lunga data, abituati a partecipare alle community, tuttavia è un materiale generale per i nuovi arrivati poiché l'ecosistema che circonda lo Yocto Project è uno dei più grandi benefici del progetto. Consolida i passaggi iniziali necessari per iniziare a partecipare alla community dello Yocto Project, quali canali di comunicazione sono comunemente utilizzati e le linee guida generali sulle buone pratiche durante la comunicazione tramite questi canali.

Si considera lo Yocto Project come un progetto molto eccitante e un potente framework utilizzabile in numerose aziende e progetti. Anche se gli aspetti tecnici più profondi relativi allo Yocto Project non sono stati trattati in questo libro, aiuta a capire le ragioni che ne guidano l'adozione e fornisce un background sufficiente in modo che non sembri più una scatola nera.