

## RED HAT FORUMS

# Openshift & Containerized Applications to create a journey of success in Financial Services Industry

Andrea Casini  
Datacenter & Automation Team Leader  
20 Novembre 2019 - ROMA  
3 Dicembre 2019 - MILANO

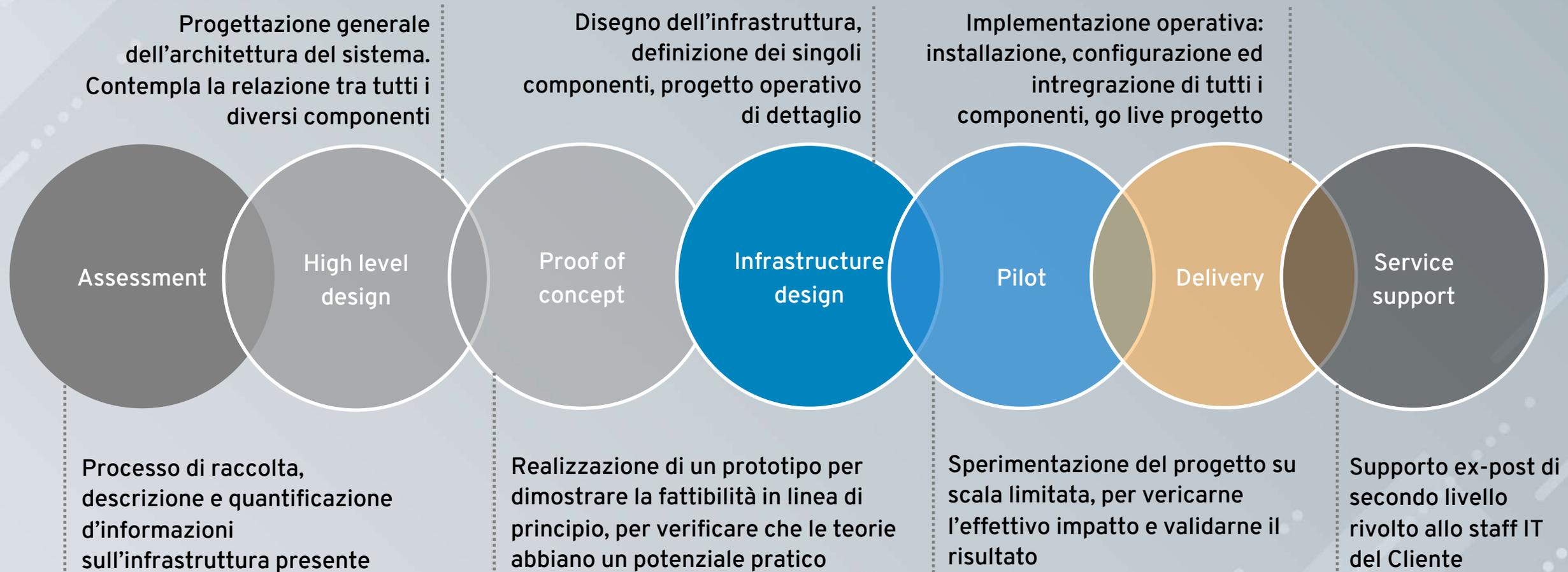
# Chi è SINTHERA?

# Azienda di consulenza innovativa e dinamica

- Le forti e consolidate competenze di consulenza, supporto operativo e formazione permettono di affrontare **progetti complessi**, integrando le **eccellenze tecnologiche** in architetture che soddisfino sempre le **esigenze del Cliente**, la **salvaguardia degli investimenti** e **l'innovazione**.
- **Ambiti: Data Center, Networking, Virtualizzazione e Servizi Cloud**, con focus su soluzioni **Software-Defined Data Center e Cloud-oriented**. Il tutto con una attenzione particolare alla **Sicurezza**.
- **Volume d'affari: 11 milioni di euro**
- **Risorse: 30 addetti**
- **Territorio: 3 sedi operative (Udine, Padova, Reggio Emilia)**
- **Mercati di riferimento: Banking , Services, Industrial**



# La Metodologia



# Delivery Teams

- L'attività di delivery dei progetti è erogata attraverso Teams composti da **consulenti specializzati**.
- Il Team Leader organizza il gruppo di lavoro, supervisiona i progetti e traguarda gli obiettivi concordati con i Clienti.
- 4 Teams:
  - **Security**
  - **Network**
  - **Datacenter & Automation**
  - **Application & Workspace**

# DevOps

# Nuove sfide? Sì e tante

- Time to market:
  - Tradizionale → mesi
  - Richieste → cicli di rilascio settimane/giorni
- Fornire tempi di risposta:
  - aderenti alle esigenze di business
  - prevedibili e deterministici
  - più rapidi

# Conseguenze delle nuove richieste del mercato...

- Maggiore utilizzo ed ottimizzazione delle risorse esistenti
- Aumento dei workload (es. container)
- ... con il rischio di perdere la governance :-)

# Approccio tradizionale: le note dolenti

- Rate di fallimento → alto
- Passaggio fra gli ambienti → manuale, spesso fra sistemi non identici e non standardizzati
- Problemi tecnici e di responsabilità nel passaggio in produzione, rimpallo responsabilità fra team Developers e Operations:
  - "*è il tuo codice*"
  - "*è la tua VM*"
  - "*hai cambiato le librerie*"
  - ....

# E' possibile mitigare questi aspetti?

- Adozione di una Metodologia
- Strumenti di Automazione
- Piattaforme di Orchestrazione

# DevOps: come ci può aiutare

- E' innanzitutto un **MINDSET**
- Set di **best practices** e di **cambiamenti culturali** supportati da **strumenti automatici** e processi di **Lean Management**, che consente di automatizzare il rilascio del software rispetto alla sua catena di produzione, permettendo alle organizzazioni di poter contare su software e applicazioni di qualità superiore in modo estremamente più rapido, per accontentare i clienti nel modo migliore e più rapidamente.
- Nascita di un unico team con competenze di sviluppo e operations che gestisce tutte le fasi in autonomia fino al rilascio

# DevOps: come si declina nella realtà

- Team distinti si cross-allineano → MetaTeam
- Comunicazione → collaborativa, trasparente e su base continuativa
- Definizione di un processo, coinvolgente ruoli multidisciplinari, con alti livelli di automazione

# Quali risultati con la nuova metodologia?

- Tempistiche passaggio in produzione → drasticamente ridotte
- Rate di change al codice → altissimo rispetto approccio tradizionale
- Failure rate → fortemente contenuto
- Passaggio fra i vari ambienti (eccetto la produzione) → fatto in autonomia dagli sviluppatori
- Passaggio in produzione diventa il lancio di un processo automatico, ripetitivo e deterministico (es. pipeline)
- Configurazioni (applicazioni ed ambiente) → esternalizzate e versionate

# Esiste un percorso strutturato?

- Decisione sulle tecnologie e sui processi da adottare:
  - Approccio metodologico DevOps
  - Disaccoppiamento stream di lavoro tra Devs ed Operations
  - Adozione di strumenti per la gestione dei progetti secondo queste metodologie

# Le persone prima di tutto!

- Incontri periodici (es. settimanali) fra le figure coinvolte per definizione degli standard (Sicurezza, Ops, Dev, Architetture software):
  - Condivisione delle scelte metodologiche
  - Cambio approccio culturale → percorso di trasformazione metodologie di lavoro

# Caso reale

- I primi mesi → focus su:
  - Implementazione piattaforma
  - Curva di apprendimento
  - Inserimento primo grado di automazione
  - Creazione deployment e oggetti di piattaforma

# Si inizia...

- Vengono containerizzate applicazioni di front-end esistenti con minima reingegnerizzazione
- Middleware
- Stateless Workloads

# ...si continua...

- Le nuove applicazioni vengono richieste secondo architetture di microservizi
- Nel frattempo aumentano le iniziative che afferiscono alla piattaforma e quindi alle metodologie relative
- altri team comprendono i vantaggi e vogliono abbracciare il cambio culturale

# ...si arriva a regime

- Affinamento dell'automazione
- Aumento della confidenza tecnologica
- Per aderire a queste metodologie e tecnologie la parte organizzativa arriva ad uno stato di maturazione elevato, basato sull'esperienze

# Esempio di una nuova organizzazione a regime

- Produzione → gestita dagli Ops
- Gli altri ambienti (Dev, Test, UAT,...) → di competenza degli sviluppatori
- I "tenant" (e relative risorse) → creati e messi a disposizione dai team di Ops

# Key takeaways

- Velocizzazione passaggio in produzione
- Aumento success rate deployment
- Chiara delimitazione responsabilità
- Indipendenza operativa fra team Ops e Dev, ma consistenza del processo comune e complessivo
- Condivisione di un metodo e di una vision
- Maggiore agilità
- Riduzione errori e tempistiche tramite forte connotazione di automazione

# Perché Openshift?

- Strumenti di integrazione pipeline CI/CD
- Semplicità di implementazione ed utilizzo
- Velocità di partenza, riduzione tempo per ottenere risultato
- Supporto da parte del vendor Red Hat

# Next step?

## La persistenza del dato

- Il 70% delle applicazioni di classe enterprise necessita di scrivere dati
- Adozione di nuove proposizioni software che aderiscono ad architetture di microservizi e adatte alla gestione sotto forma di container
- Red Hat tramite Operator Framework è in grado di agevolare l'adozione di StatefulSet semplificando il deployment e il lifecycle management
- Openshift completa il supporto a queste iniziative con una proposizione tecnologica di Software Defined Storage integrata con la piattaforma che oltre a fornire lo storage di backend porta anche la visibilità in termini di metriche di performance (Red Hat OpenShift Container Storage)

# RED HAT FORUMS

# THANK YOU



[linkedin.com/company/Red-Hat](https://www.linkedin.com/company/Red-Hat)



[facebook.com/RedHatinc](https://www.facebook.com/RedHatinc)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[twitter.com/RedHat](https://twitter.com/RedHat)