Re-engineering Engineering: from a cathedral to a bazaar?

Vinod Khosla

Sept 2000
Environment: “Change as a Process”

- Business model evolution everyday!
- Infrastructure renovation
- Systems evolution
- Strategy evolution
• Specialization & complexity of technology
• Decision-making: top down or bottom up?
• The role of the “fringe” employee
• Nuances as pitfalls
• Horizontal & vertical communication & cooperation - not top down
• Information based, dynamic decision making
The CIO’s Issues
CIO’s Issues

- The problem of legacy - systems, people, ...
- **Skills shortage**
- **Re-engineering the enterprise for technology based competition/strategy**
- Intranets & extranets among islands of information/systems
- **Dynamic information architecture** vs. static databases (“enterprise models”)
- **Real time corporation** & future of software
- New application proliferation
CIO’s Issues: Legacy Engineering

• Optimization for what
  – Cost
  – Performance
  – Reliability

• Systems

• Business Process
CIO’s Issues: Change Management

- Old databases
- Old systems
- Legacy logic
- C/S architectures
- New applications
- New users
- New “internet” environment
- Multi-architecture systems
CIO’s Issues: Skills Shortage

- **Complexity increasing exponentially**
  - More systems
  - More applications
  - More devices

- **Rapid change**
  - Faster versions
  - New requirements

- **Human capital**
  - Linear growth of supply
  - Outflow from MIS
CIO’s Issues: Engineering Methodology

- Evolvability
- Specialization
- Experimentation
- Change isolation
- Diversity
- Connectivity oriented
- Best of breed oriented
- Standards
The Road Ahead
**Road Ahead: “New” Goals**

- **Complexity thru federation** NOT integration
- **Adaptability & evolvability**
- **Configurable** NOT customized
- **Modularity** – “micro” open systems model
- **Personalization**
- **Application interoperability, unified UI**
- **Dramatically new management systems**
Road Ahead: A “new” Reliability

- The shuttle Challenger: designed not to fail
- Biological systems: designed to fail gracefully
- Complex systems: “evolutionary approach”
- 24/7 mission critical systems (Routers vs. phone network)
So what does this have to do with 2014?

... on to SDN’s & more
FW rule
allow web7
vlan 225-318
allow tcp 22
deny all
allow tcp 80
allow any any
tcp 22
FW rule allow web7
allow tcp 80,443
allow tcp 8080
vlan 10-12 on eth2
allow from 10.4.3.22/28
allow any any tcp 22
vlan 480-490
allow tcp 480-490
! through FW 3 only
Configuration & provisioning...

**1996**

```
Router> enable
Router# configure terminal
Router(config)# enable secret cisco
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3
Router(config)# interface ethernet0
Router(config-if)# ip address 10.1.1.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial0
Router(config-if)# ip address 20.2.2.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
Router(config-router)# exit
Router(config)# exit
Router# copy running-config startup-config
Router# disable
Router>
```

**Terminal Protocol:** Telnet

**2013**

```
Router> enable
Router# configure terminal
Router(config)# enable secret cisco
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3
Router(config)# interface ethernet0
Router(config-if)# ip address 10.1.1.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial0
Router(config-if)# ip address 20.2.2.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
Router(config-router)# exit
Router(config)# exit
Router# copy running-config startup-config
Router# disable
Router>
```

**Terminal Protocol:** SSH
16 racks, 500 apps, 1500 vlans...

...34 mgmt consoles, 80k lines of config

Configuration & provisioning...
Configuration & provisioning...

- logical sub-nets vs. physically networks
- North-south vs east-west traffic
- New types of Workloads: Dev, VDI, Big Data
- security
  - service chaining
  - Multi-path routing
  - Fault isolation
  - New (mobile) devices

- Static Network
Complexity →

Limit of human understanding

Time, scale, performance →
2014: Problems

#1 OPEX cost reduction: autonomic

#2 Complex provisioning of applications and services: virtualized data center

#3 Visibility and diagnostics: analytics & “autonomics”

#4 Hardware network appliances: network function virtualization

#5 Network is brittle, Fault isolation is hard: self-healing

#6 Management thru NSX, System Center & OpenStack: datacenter OS

#7 Vendor lock: modularity & open systems
SD(X) facilitates...

- reducing talent shortage
- dynamic needs of virtualized data center
- self-healing basis
- datacenter OS (thru virtualized compute, storage, network)
- autonomic management basis (mostly software changes)
- NFV: expanding scope of devices
Software Defined Networks

• A stronger intellectual foundation to networking

• Helps define the right abstractions

• Formally verify correct network behavior

• Identify bugs, then systematically track down their root cause

“How SDNs will tame networks”  Nick McKeown, Stanford University
Re-engineering Engineering Methodology
Re-engineering Engineering Methodology

• Evolvability: unpredictable workloads (Dev, VDI, Big Data…)

• Specialization (in SW)

• Change & fault isolation

• Experimentation
Re-engineering Engineering Methodology

- Abstractions at multiple levels (modularity)
- High rate of evolution “design”
- Specialization / dynamic specialization
- Obsolescence strategy
- Autonomic
Re-engineering Engineering

Optimize for performance

Optimize for flexibility

Time

Change
The holy “COW”: OPEX

...or AUTONOMIC replacing “human judgment”

...or PACMAN eating away at innovation?
Security: Complexity begets hackability

- Windows
- Financial crisis
- Brain: Candy Crush, Adtech
Do you Google?
PAAS → A MODEL FOR APPLICATIONS

Dynamic applications (web, big data) demand Infrastructure & Platform exposed as-a-service.

- Compute
- Storage
- Database

Google App Engine

- Abstracted Services for Apps (DBs, Messaging ..)
- Dynamic, Distributed apps
- new business models (as-a-service)

Platforms must be open/interoperable, scale-out/elastic, converged, multi-tenant, automated.
Management or Datacenter OS?
monolithic static cluster provisioning (OpenStack) vs. dynamic distributed resource optimization (Omega/Mesos)

• Scale cluster/data center size
• Respond to changing requirements
• Increase rate of new feature deployment
• Increased efficiency and utilization
• Add new policies and specialized implementations
• Make decisions that require state of the entire cluster
Network Function Virtualization (NFV)

... hardware-based appliances rapidly reach end of life

... integrate-deploy cycle inhibiting roll out of new revenue services

... hardware lifecycles are becoming shorter constraining innovation

... specialized equipment increase need for specialized talent locally
Everything as a Service (XaaS) ... a building block for more complex services

... a well-defined Function with programmatic API

... scalable, elastic, and resilient

... built out of unreliable components (self-healing)

... implemented as Network of VMs
Future of Computing: 1,2,3...

Step 1: “Drop” it into cloud

Step 2: It becomes a service

Step 3: Millions of end-users sign up

ONRC/ON.Lab Overview
Programmable Virtual Networks On Demand: SDN Control Software as SaaS with Service Composition
SDN is the beginning of the software era

- Layering/componentization => **enable innovation**
- Compute/storage => extract simple abstractions from complexity
- Networking => “Mastering complexity” (past vs. SDN future)
- Abstraction & modularity key to **extracting simplicity**
- Three key abstractions: **distribution, forwarding, configuration**

The Future of Networking, and the past of protocols by Scott Shenker
[https://www.youtube.com/watch?v=YHeyuD89n1Y](https://www.youtube.com/watch?v=YHeyuD89n1Y)
Autonomous/adaptive/aware, self-healing, self-optimizing computing (compute, storage, networking)

SDN, SDS, SDC, SDDC, NFV, PaaS, XaaS,... part of a larger context of “needs” not “technologies” and “new abstracted/modular how’s” to “humans out of the loop” near real time, complex self management, dynamic, evolvable systems
other thoughts

transactions per 100ms: human-scale interactivity
modularity: “fractal systems” (self-similar, self-optimizing, self-healing)
fail gracefully: making predictable whole from unpredictable parts
SLO’s (Service Level Objectives): machine equivalent to SLAs
evolvable systems: meaningful selection pressure
new (mobile) devices: “pervasive computing”
“obsolescence strategy” à “selection strategy”?"
New Area: “Virtual Computer”

A Computer Distributed Over the Internet

- Scalability of hardware - add & delete
- Self management
- Geographic distribution
- Load balancing, caching, COS, … services
- “Network operating system” for the IBASE
Evolvable Systems (Sharky)

- Only solutions that produce partial results when partially implemented are evolvable.
- What is, is wrong.
- Evolution is cleverer than you are.

Centrally designed protocols start out strong and improve logarithmically… evolvable protocols start up weak and improve exponentially.
Linux: Cathedral and the Bazaar
Given the essential ingredients of evolution ... any system, natural or artificial, can evolve into a complex design through incremental changes explored in parallel.
Personal Views: Development Mechanisms

- Modular development
- Successive refinement
- Aggressive peer review
- Forced “Architecture, Architecture, Architecture”
Personal Views: Methodology Ultimate “Open System”

• Origin of “open systems” circa 1982

• Methodologically forced openness

• Methodologically forced modularity

• Methodologically forced adaptability
a 1996 story: an ATM internet?

... evolvability & modularity win because of enabled innovation

... you can “fit in” or change the future
Weather Forecast

Rate of change will accelerate

Innovation, opportunities & entrepreneurship will thrive

Fun & fortunes will be in abundance

Adaptability, agility & momentum will be the key to success!
There’s change and then there is change!

“...every strategic inflection point [is] characterized by a ‘10X’ change ...”

“There’s wind and then there is a typhoon, there are waves and then there’s a tsunami”

- Andy Grove
Comments?
Resumes?
Business Plans?

Vinod Khosla
Mar 2014
vk@khoslaventures.com
Reading:

- The Cathedral and the Bazaar (Eric Raymond)
- In Praise of Evolvable Systems (Clay Shirky)
- The Circus Midget and the Fossilized Dinosaur Turd (Martin Hock)
- Linux: A Bazaar at the Edge of Chaos (Ko Kusabara)
Software defined networks, storage, compute (PaaS)

Hybridized public & public clouds

Network Function Virtualization

Security beyond a “feature”

Applications as a Service

Autonomic systems

.... a new computer architecture beyond client-server derivatives
APPS NEED ELASTIC ARCHITECTURES

- IaaS / Paas
- Open-source, standards based
- Scale-out architecture
- Policy-based automation
- Real-time Analytics, Insight

Dynamic, Big-data applications

- Continuous Infra Feedback
- Resource Orchestration
- Analytics
- Compute & Storage
- Network Orchestration
- Network Services Orchestration
- Virtualized Network
- Events, Logs, Statistics
- Policy & Security Framework
network virtualization viewed as an extension of server virtualization

Open & Interoperable

Abstractions modularity

Datacenter OS

Cloud Networks (2014+)

AI

Distributed & scale-out, bi-directional control, analytics

“SDN”

Distributed & scale-out, bi-directional control, analytics

Static & Configured

Openflow, separation of control & data plane

Virtualization

“SDN”

Datacenter OS

Cloud Networks (2014+)

Storage

Compute

Virtualization

Networks

Storage

Dynamic with Real-time feedback loop

Traditional Networks (1990’s, 2000’s)

Network virtualization viewed as an extension of server virtualization

Proprietary & Silo’ed

AI