

# Beyond Virtualization

---

Michael Connly, CTO – UnitedHealth Group

## Cloud Computing – Key Enabler: Infrastructure As a Service

---

- The key characteristics of ***Infrastructure as a service***]:
  - **Resources delivered as a service** including servers, network equipment, memory, CPU, disk space, data center facilities,
  - **Dynamic scaling** of infrastructure which scales up and down based on application resource needs
  - **Variable cost** service using fixed prices per resource component
  - **Multiple tenants** typically coexist on the same infrastructure resources
  - **Enterprise grade** infrastructure allows mid-size companies to benefit from the aggregate compute resource pools

## UHG - How it was

---

### End of 2004

- 90% of Infrastructure built on a to-order, project-by project basis with capital owned by project.
  - Little standardization.
  - Server utilization: 8%.
  - Infrastructure staff grew linearly with application development staff.
  - All projects initiation and capacity expansions were at the mercy of a procurement cycle.
  - Little leverage for tool implementation.
- 

3

## UHG: Setting the Stage -- Managed Services (Infrastructure as a Service)

---

### Today

- 90% of infrastructure shared across projects.
- Over 500 applications using shared servers.
- Capital owned by the infrastructure organization.
- Highly standardized.
- Infrastructure organization chooses hardware and technology--- provisions capacity on a real time basis.
- Server utilization of 60%.
- 60 days forward inventory on hand.
- Much faster deliver of infrastructure.
- Developer enablement tools can be implemented, as the investment can be leveraged.

Highly Virtualized

---

4

## But Virtualization Is Only A Starting Point

### ■ Current State

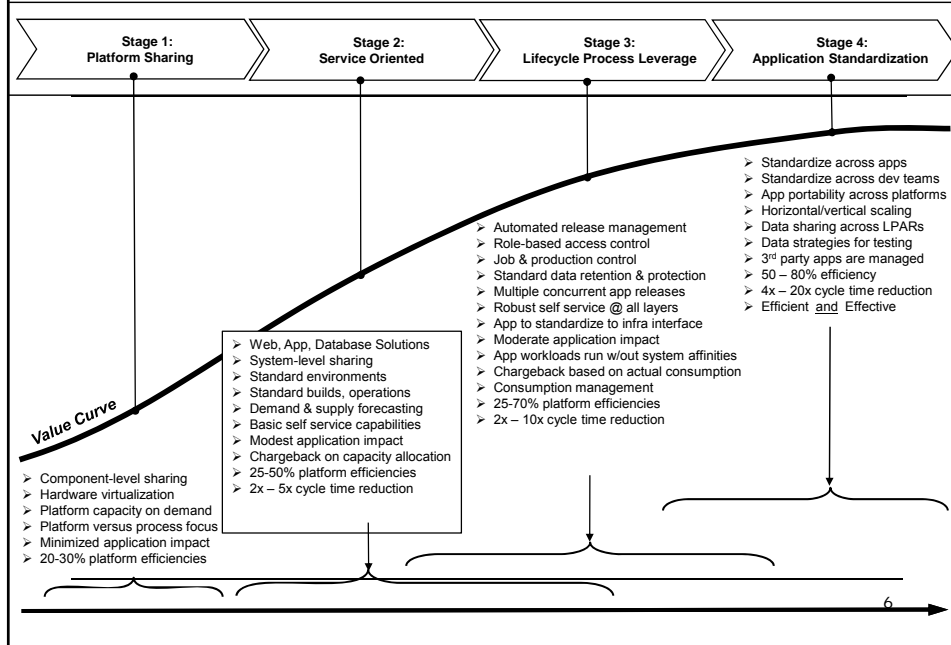
- Technology has been standardized (Managed Services) but application development highly individualized, producing:
  - Variability – significant differences in how similar solutions are implemented
  - Inability to Audit – cannot verify use of best practices
- Limited Leverage – inability to access experts on huge number of partner areas that will be required to deliver a solution

### ■ Results in Increased:

- Cost – larger application footprint, duplicative functions
- Complexity – related applications managed very differently
- Rigidity – solutions difficult to change / adapt
- Cycle Time – custom infrastructure configurations, custom development
- Defects – no “best practices” available to avoid repeating easily-avoidable mistakes
- Risk – investment may be wasted if successful solution is not delivered

5

## UHG Managed Services Capability Maturity Model



## Managed Services –Stage IV

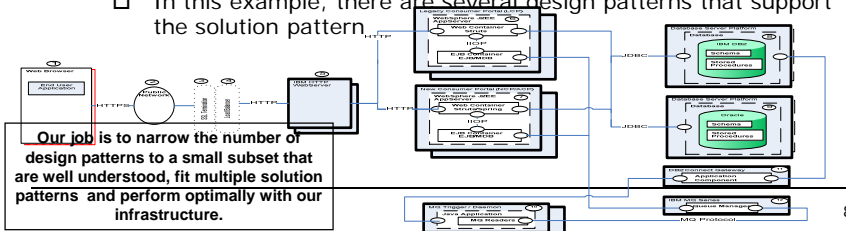
- ❑ Address next source of variation i.e. applications
- ❑ Provide integrated solutions: (virtualized infrastructure + application frameworks)
- ❑ Use patterns to decide on what to integrate
- ❑ Align requirements process with design

Sophisticated Clouds!

7

## What are Patterns

- ❑ Solution Patterns
  - A solution pattern is commonly occurring solution that is generally used to support certain kinds of business needs
    - ❑ Example: Self Service Web application
- ❑ Design patterns
  - a design pattern is a general reusable pattern to a commonly occurring problem in software design. It is a description or template for how to solve a problem that can be used in many different situations
    - ❑ In this example, there are several design patterns that support the solution pattern.



## Technology Solution Patterns

---

- IT solutions can be categorized based on patterns of business requirements and “performance” needs
  - While a **Business Need** will normally generate multiple **Business Problems, Business Solutions** (where technology would participate) will fall into only these Groups:
    - **User Application** to be Built/Modified, **or**
    - **Middleware Application** to be Built/Modified, **or**
    - **Enterprise Repository** to be Built/Modified

The infrastructure and basic application frameworks for these IT solutions can be bundled and provided on demand!

9

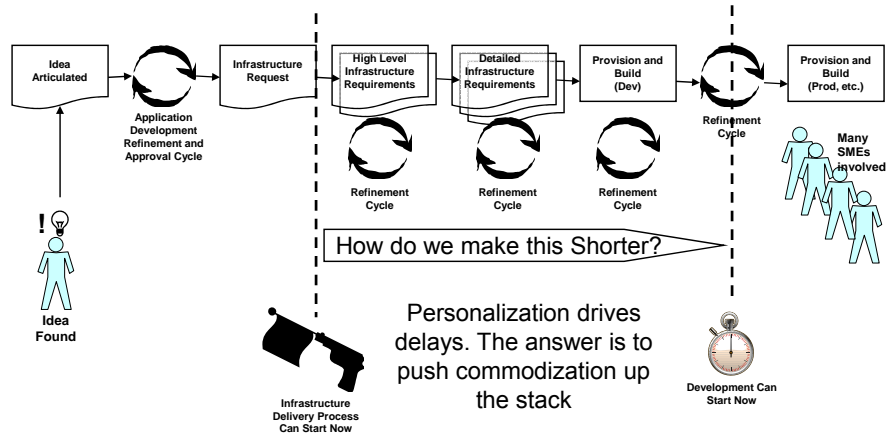
## User Application Patterns - Example

---

- Example:
  - Self Service Web application
    - Requested frequently (200+ infrastructure requests over 12 months)
    - Differences – business logic, and performance characteristics (Concurrent users, through put, load etc)
      - These fall in 2 categories – Medium, Large patterns of implementation
    - Patterns Implementation will provide pre-configured, blank functioning application (“hello world” web app) ready to be customized with business logic
      - Other design patterns will provide “how to” information on how to write optimal, well performing code

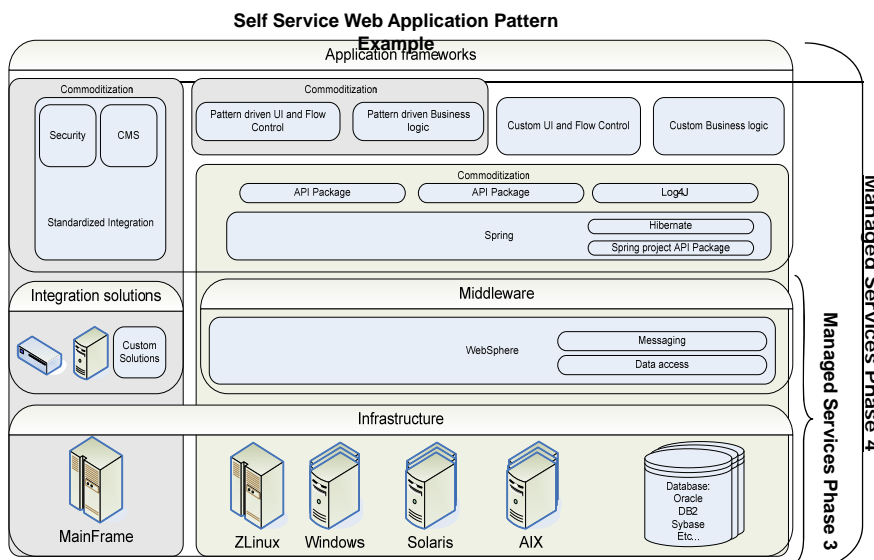
10

# Current: Infrastructure Delivery



11

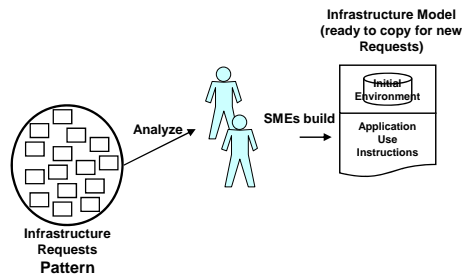
# Pushing Commoditization up the Stack!



**Preconfigured for Small, Medium, Large – no personalization delay!**

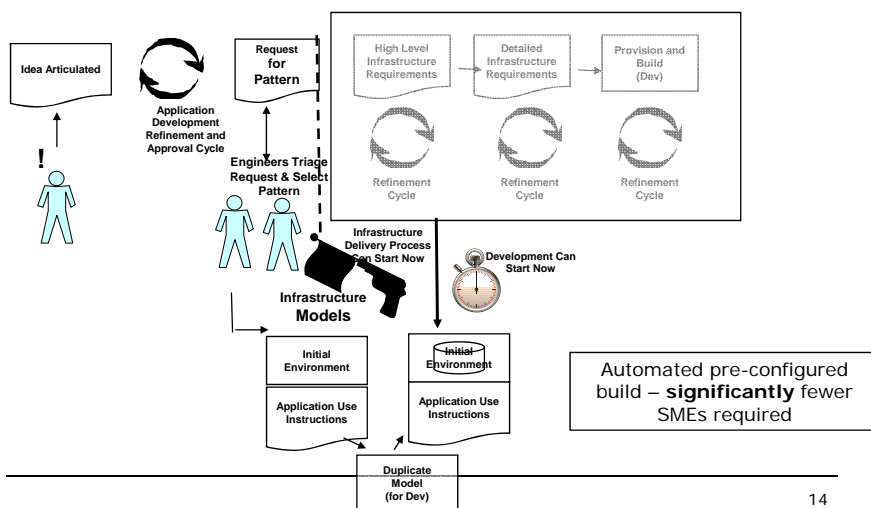
12

# Infrastructure Patterns – Template creation



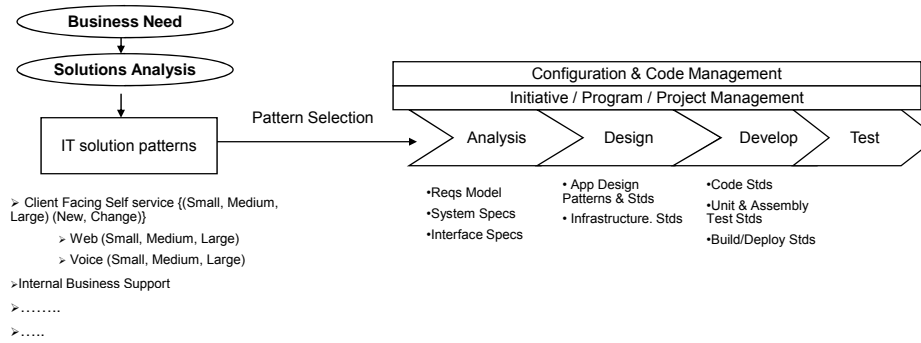
13

# Patterns Driven Infrastructure Delivery



14

# Pattern Driven Solutions Delivery



15

# What do Developers Get?

- Standards Environments with Good Documentation
- Number of environments to support concurrent development predetermined
  - Support services such as Source Code Management, Build and deploy services provided as part of the "package".
- Limited Number of Patterns
  - Triage in Requirements to Standard Pattern (where possible)
  - Clear Characteristics and Restrictions on Pattern Usage
  - Once Pattern Chosen team Stays within Usage Specifications
  - New Extensions to Patterns are Cycled back into Standard
- Development Community is Active Participant
  - Rich Application Development Toolset
  - Thought Leaders Engaged

16

## Pattern Driven Solutions Delivery: Benefits for the Enterprise

---

- Application Frameworks and Infrastructure Constraints rationalized to deliver
  - Reduced Infrastructure Cost
    - Minimum/Optimal Number of Development & Test Environments
    - Ability to leverage enhanced capabilities of virtualization
    - Std environments – faster delivery
    - Std environments – quicker problem sectionalization process
  - Reduced development cost
    - Quick Start
    - Reduce Cost and Complexity
      - Streamline Application development lifecycle by providing standard toolkits and services
    - Improve Flexibility and shorten Cycle Times
      - Offer pre-engineered solutions for common problems
    - Eliminate Defects
      - Enable continuous improvement to pre-engineered solutions
    - Mitigate Risks
      - Common Solutions allow predictable results
    - Leverage Practitioner expertise
      - Create a forum to enable asynchronous, distributed, controlled collaboration that remains current through continued participation

---

17

## Patterns Based Solutions Delivery - Challenges

---

- Build the wrong thing faster
  - Ensuring “right” level of commodization/standardization
- Infrastructure provisioning could go out of control
- NIH at team level
- Environment is right, problems are in the code
- If you change the infrastructure to fix an Application or DB problem, game’s over!

Understanding Requirements Patterns and providing proper Governance is key!

---

18

---

## Build the Right Stuff

---

19

## Common Problems with Requirements

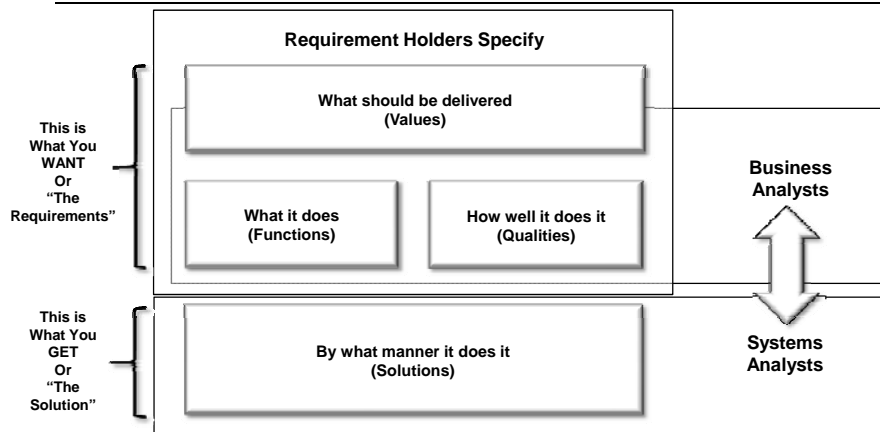
- Failure to include all **relevant requirement holders**
- Failure to document **clear business values** provided by the requirement
- Failure to specify the **critical product quality requirements** (such as scalability or reliability)
- Failure to include **objective measures** for assuring that a requirement is met
- Documenting design** rather than requirement
- Requirements **repeated in different ways** across multiple documents
- Failure to allow subsequent process steps the **flexibility to choose best** (or better) ways to produce solutions

It starts with requirements. Working toward a better understanding of the problem by analyzing and documenting the parameters of success. Historically, we've faced a number of challenges in gathering and recording useful requirements. Addressing these challenges will make us better at our job.

Any use, copying or distribution without written permission from UnitedHealth Group is prohibited.

20

## A Model for Requirements and Solutions per Gilb



Tom Gilb ([www.gilb.com](http://www.gilb.com)) offers this model for defining requirements. Business Analysts and Systems Analysts, working together with the Requirement Holders, define the parameters of success for the project.

21

## Typical "Before" Requirement – Service Calls

3	2	Request Status	Expand the information on the status request page for users to include: current status (need more than just "Suspended" or "Completed"), the name of the approver currently reviewing the request and the name of the next approver	High	
---	---	----------------	---	------	--

22

## Example: Service Call Requirement, After

### Request Service Calls

<b>Last Change Date</b>	4/30/2007
<b>Owner</b>	LTG
<b>Status</b>	Initial Draft
<b>Stakeholders</b>	LTG, Requestors, Service Administration

#### Comments

Requestors must be able to determine the Status of an outstanding Request at any time without relying on other individuals. This will reduce the amount and cost of Service Calls made to check Status of outstanding Requests.

#### Measurements

Number of [Service Calls] made per month checking [Status] of [Outstanding Requests]

Reference	Qualifier	Value	Source	Notes
Past	As of 2006	250	Tom Nelston	
Goal	By 2H2007	50	Tom Nelston	
Tolerable	By 1/1/2008	100	Tom Nelston	
Wish	By 1/1/2008	10	Tom Nelston	

Term	Notes
Service Call	A phone call made by an employee to the LTG service request administration support line
Status	Disposition status of a service request and related information concerning expected completion of the request
Outstanding Requests	Requests which have not yet completed processing in order to fully grant the services described by the access request

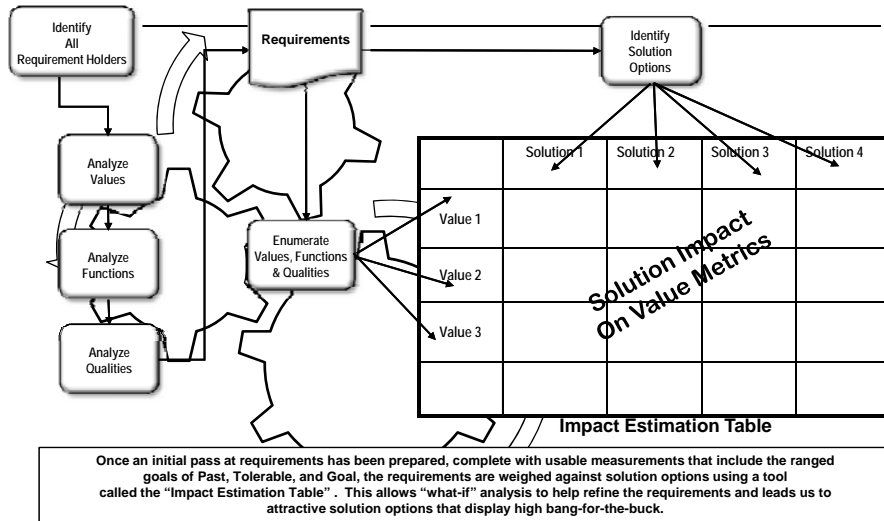
23

## Solving the Right Problem

- Impact Estimation – a systemic process for selecting solutions
- Provides understanding for how well a set of Solution Ideas satisfies a set of Requirements Goals
- Provides a mechanism for assessing
  - ▶ If a chosen Solution meets the quantified Goals
  - ▶ How one Solution, intended to satisfy a subset of Goal, impacts other critical Goals?

24

## Seeing the Whole Process: Solution Options



2  
Any use, copying or distribution without written permission from UnitedHealth Group is prohibited.

25

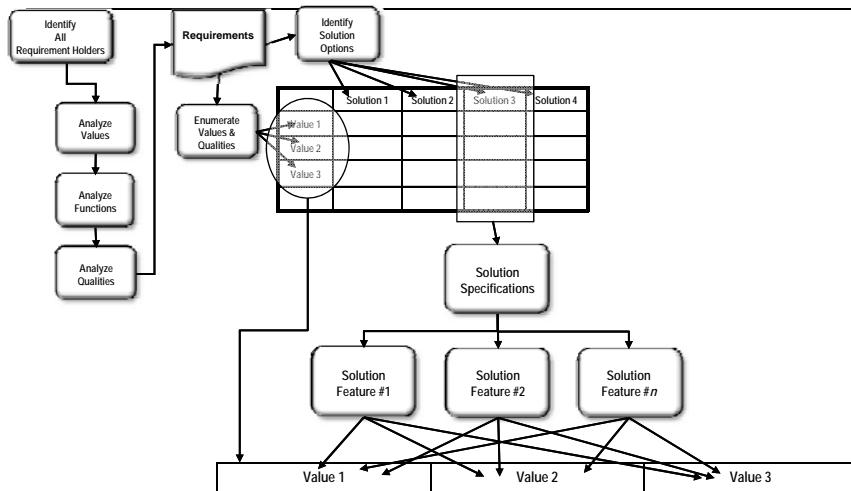
## Solutions Analysis: Improve Help Desk

Solution Ideas Values	Self Service	Add 25% more Staff	Advanced Training	New Help Desk System
Cost per Call Goal = \$ 10 Tolerable = \$11	Outcome: \$9 % of Improvement Goal: 200%	Outcome: \$14 % of Improvement Goal: -300%	Outcome: \$12 % of Improvement Goal: -100%	Outcome: \$14 % of Improvement Goal: -300%
Average speed to answer < 60 secs Goal= 90% of calls Tolerable = 70% of calls	Outcome: No Change % of Improvement Goal: 0%	Outcome: 80% % of Improvement Goal: 50%	Outcome: 75% % of Improvement Goal: 25%	Outcome: 75% % of Improvement Goal: 25%
First call Resolution Goal= 80% of calls Tolerable = 70%	Outcome: 60% % of Improvement Goal: -50%	Outcome: 70% % of Improvement Goal: 0%	Outcome: 80% % of Improvement Goal: 100%	Outcome: 80% % of Improvement Goal: 100%
Caller to HD analyst ratio Goal = 500 Tolerable = 420	Outcome: 460 % of Improvement Goal: 50%	Outcome: 400 % of Improvement Goal: -25%	Outcome: 420 % of Improvement Goal: 0%	Outcome: 500 % of Improvement Goal: 100%
Total Impact Score	200	-275	25	- 75
Solution Cost	\$1,000,000	\$1,000,000	\$200,000	\$6,000,000
Benefit/Cost ratio	200	-275	125	- 12.5

Simple Example IET: Improving the help desk based on desired parameters and offered features.

# Align Design with business quality requirements

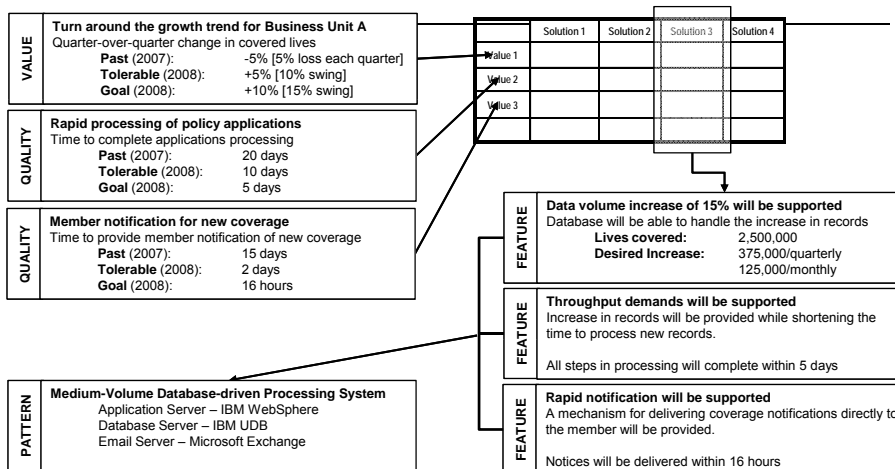
## Seeing the Whole Process: Getting To Design



2  
7

27

## Seeing the Whole Process: Simple Example



2  
8

28