Application **P**erformance **M**anagement in the **S**oftware **D**evelopement **L**ife**c**ycle

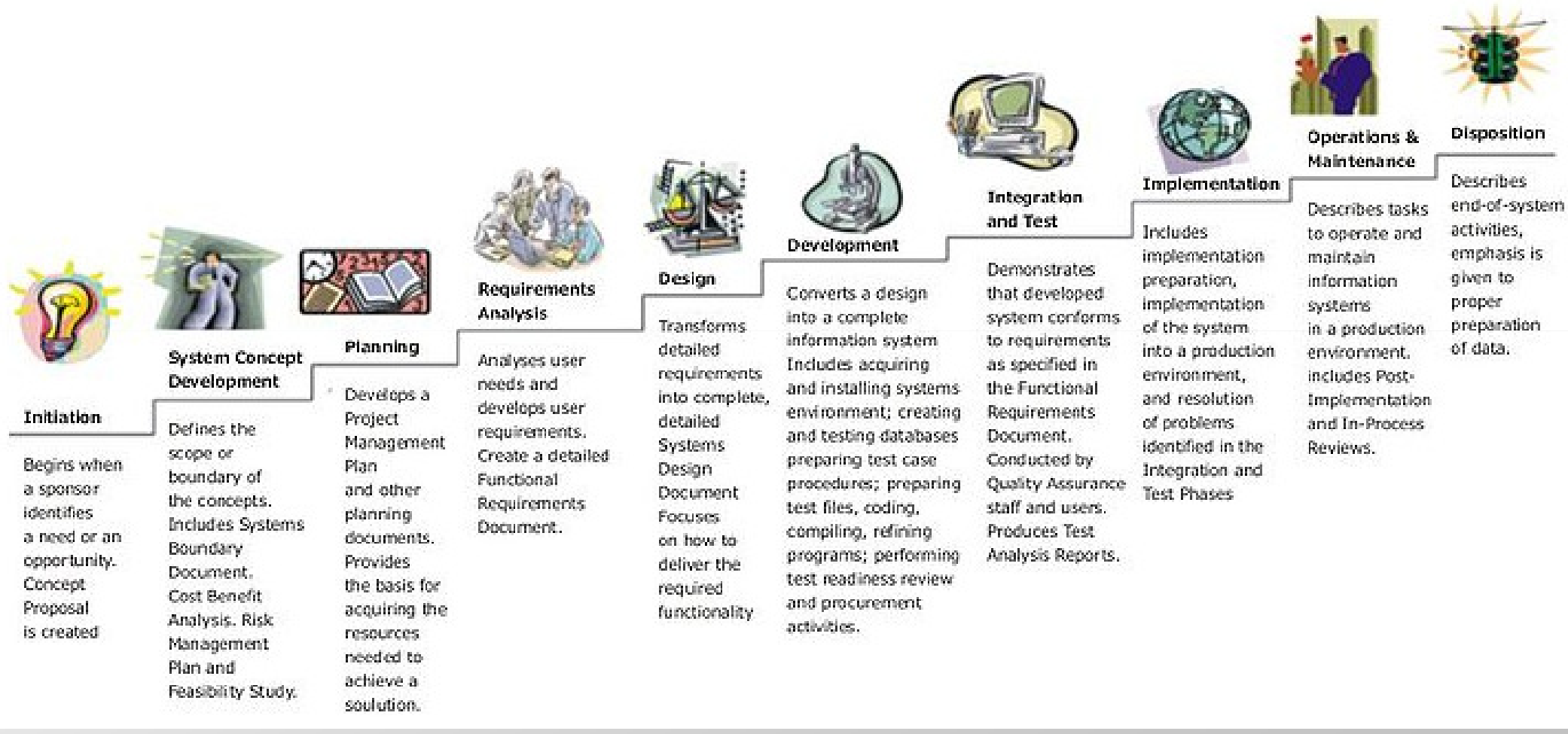
Dirk Loosen
IT Ecology
dirk@itecology.co.za

Overview of SDLC (WIKI)

SDLC is a process used by a systems analyst to develop an information system, including requirements, validation, training, and user (stakeholder) ownership ……

……a number of SDLC models have been created: "waterfall"; "fountain"; "spiral"; "build and fix"; "rapid prototyping"; "incremental"; and "synchronize and stabilize".

Overview of SDLC (WIKI)



Overview of APM (WIKI)

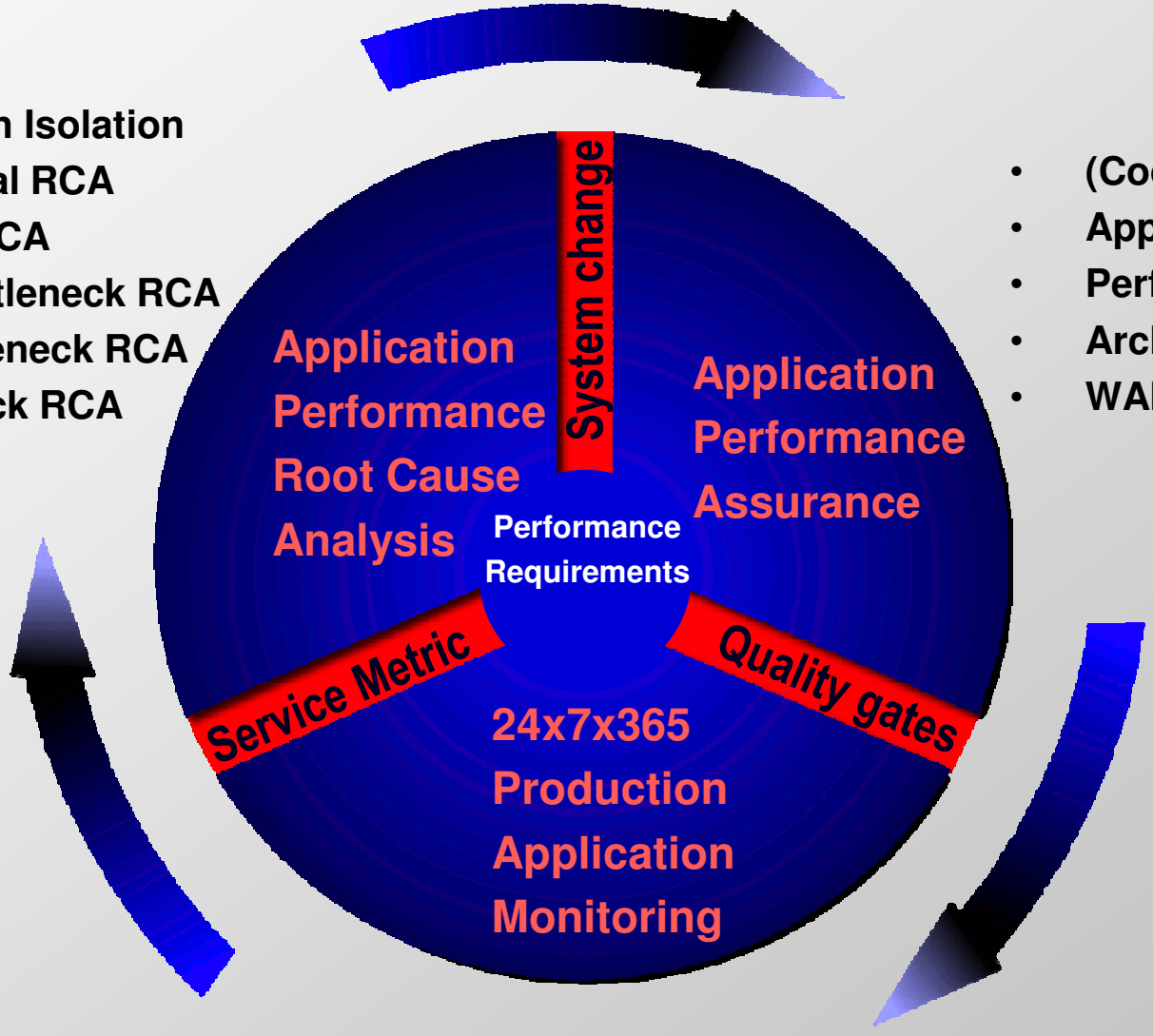
APM, refers to the discipline within systems management that focuses on monitoring and managing the performance and service availability of software applications.

APM can be defined as process and use of related IT tools to detect, diagnose, remedy and report application's performance to ensure that it meets or exceeds end-users' and businesses' expectations…….

There are two main methods by which applications performance is assessed for production applications. The first is measuring the resources used by the application, has been in use since computers have been used for business applications, and is still in use. The second is measuring the response time of applications from the perspective of the end user.

The APM process in the SDLC

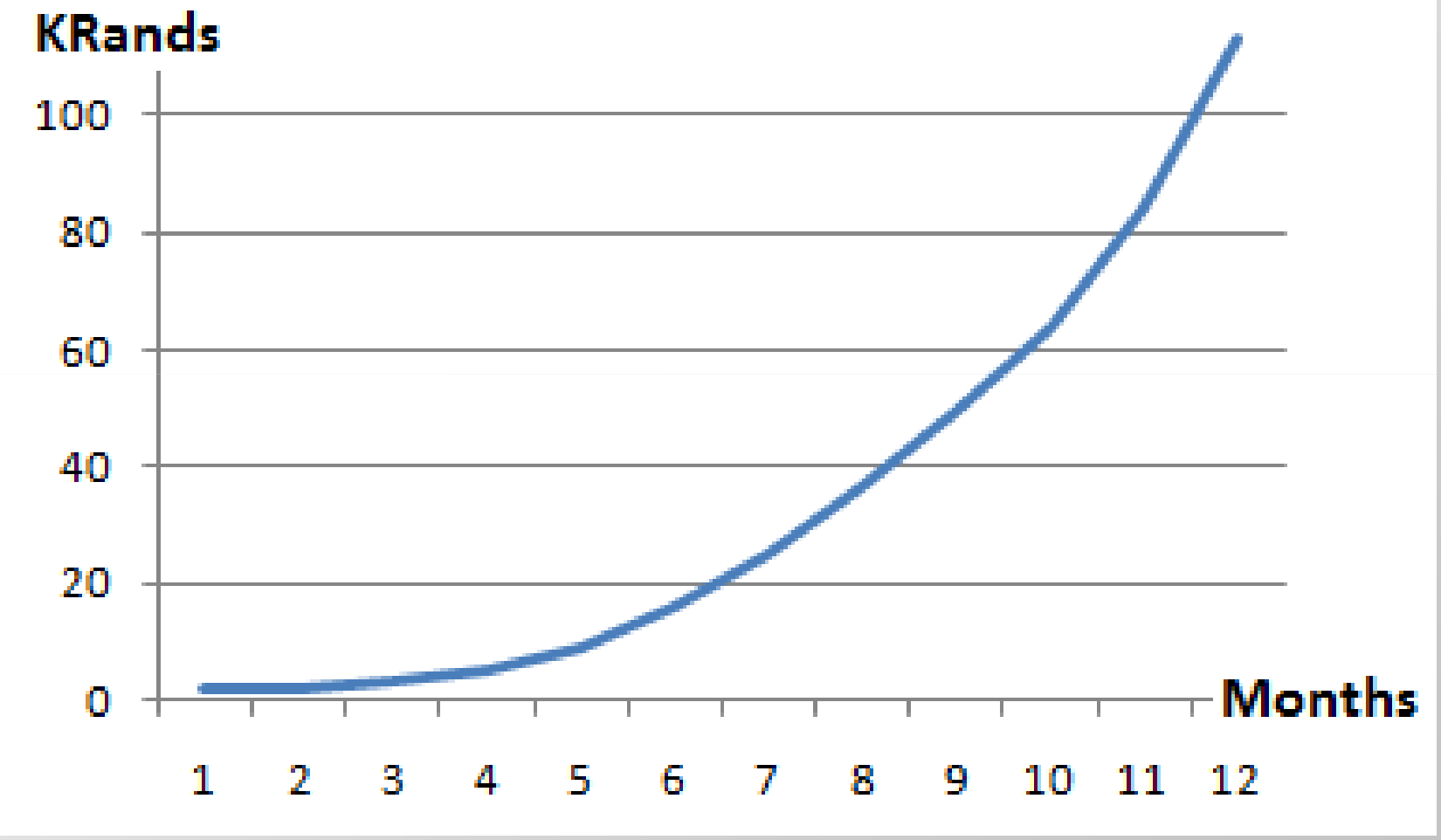
- Fault Domain Isolation
- Transactional RCA
- JAVA/.Net RCA
- Network bottleneck RCA
- Server bottleneck RCA
- DB bottleneck RCA



- (Code profiling and tuning)
- Application Profiling
- Performance Testing
- Architecture verification
- WAN Provisioning

- End User Experience Monitoring
- Application Traffic & QoS Monitoring
- Application Server Monitoring
- Baselining of normal conditions vs. usage

Why do APM early in the SDLC



A word on performance requirements

- Should be part of the concept, not considered once built
- Must be documented, agreed and signed off
- Business requirements – “This contact centre must handle a call volume of 30000 calls per month to be viable”
- Technical requirement derived – “can the system handle 30000 transactions per month and will it respond well enough not to negatively affect call duration”
- All measurements in lifecycle are taken with performance requirements in mind – what’s considered good/bad?
- Test Requirements from previous tests
- Largely dependant on type of change
- BAs need to be educated in this space

Application Performance Assurance driven by System Changes

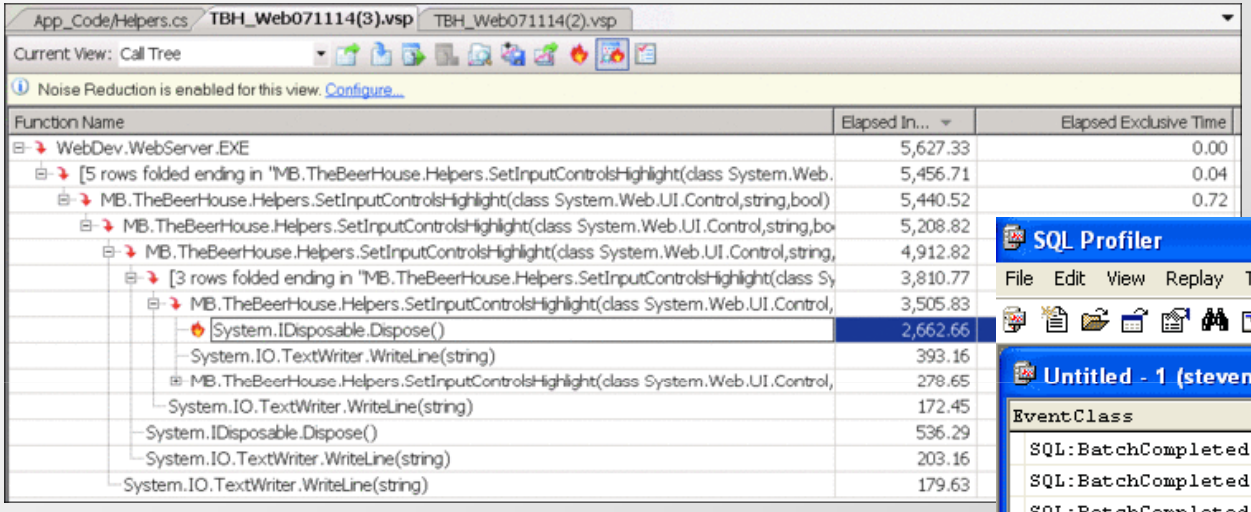
Any changes in the IT system that may have an impact on application performance should be considered. Weigh up the level of impact, the associated risk to the business (80/20 rule)

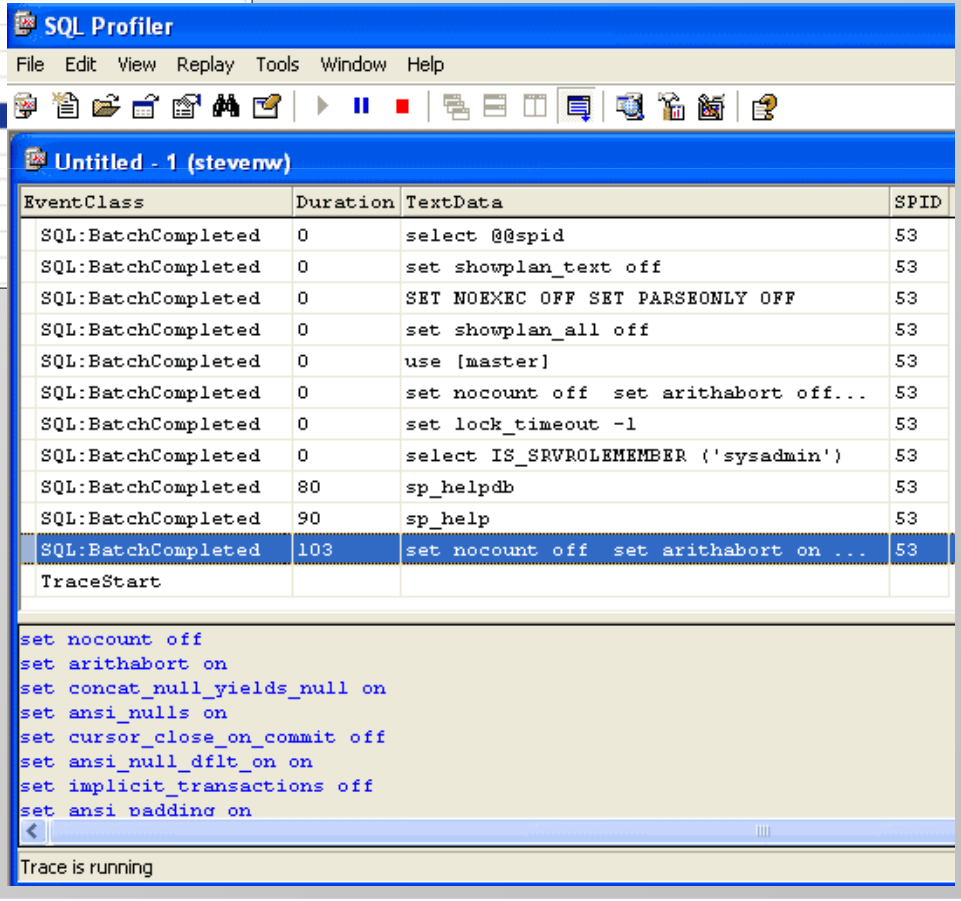
Examples:

- ***Application changes*** = New Application, additional functionality, bug-fix/patch, integration or new version of application, growth in number of users
- ***Network changes*** = Introduction of caching servers, compression, re-architecture, MPLS/QoS changes
- ***Server changes*** = server hardware change, server consolidation, virtualisation, load balancer implementation

What to test in APA phase

Code profiling and tuning – part of every developer's job



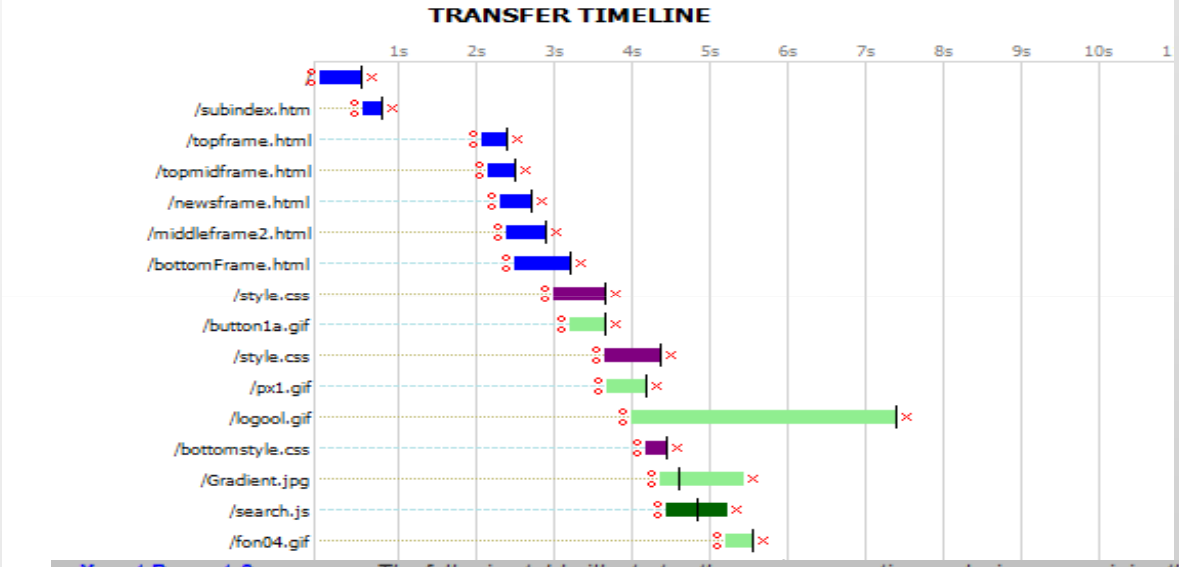


- Can I slice off CPU cycles or minimise memory footprint
- Backend – Can I make this SQL more efficient

What to test in APA phase

Application Profiling & Tuning (ASAP - once code is exe)

- Analyse packet captures of critical business process' steps



- See network footprint, latency & congestion sensitivity
- Tune where possible (back to dev? Risk vs. cost)

neXpert Report Summary - *The following table illustrates the core aggregation analysis summarizing the data captured from Fiddler.*

Step	Step Description	Response Time (s)	Round Trips	Bytes Sent (KB)	Bytes Received (KB)	Bytes Total (KB)
1		26.09	27	15.31	271.36	286.67

HTTP Response Codes - *The following table shows HTTP response codes of 300 and up (* excluding HTTP 401).*

Design Recommendations - *The following recommendations involve changes to the design and layout of the website*

Redundant Requests - *The following table shows all objects which were requested more than once in a single step.*
Static Files - *The following table shows all static files used in this scenario.*
Large Image Files - *The following table shows all image files used in this scenario which were larger than 25 KB.*

Settings Recommendations - *The following recommendations involve changes to web server settings.*

Compression - *The following table shows all steps with data transfer if GZIP compression is turned on*
Cache Headers - *The following table shows all static files which are missing either the Expires or Cache-Control header.*
Connection Header - *The following table shows all files which explicitly closed the TCP connection*

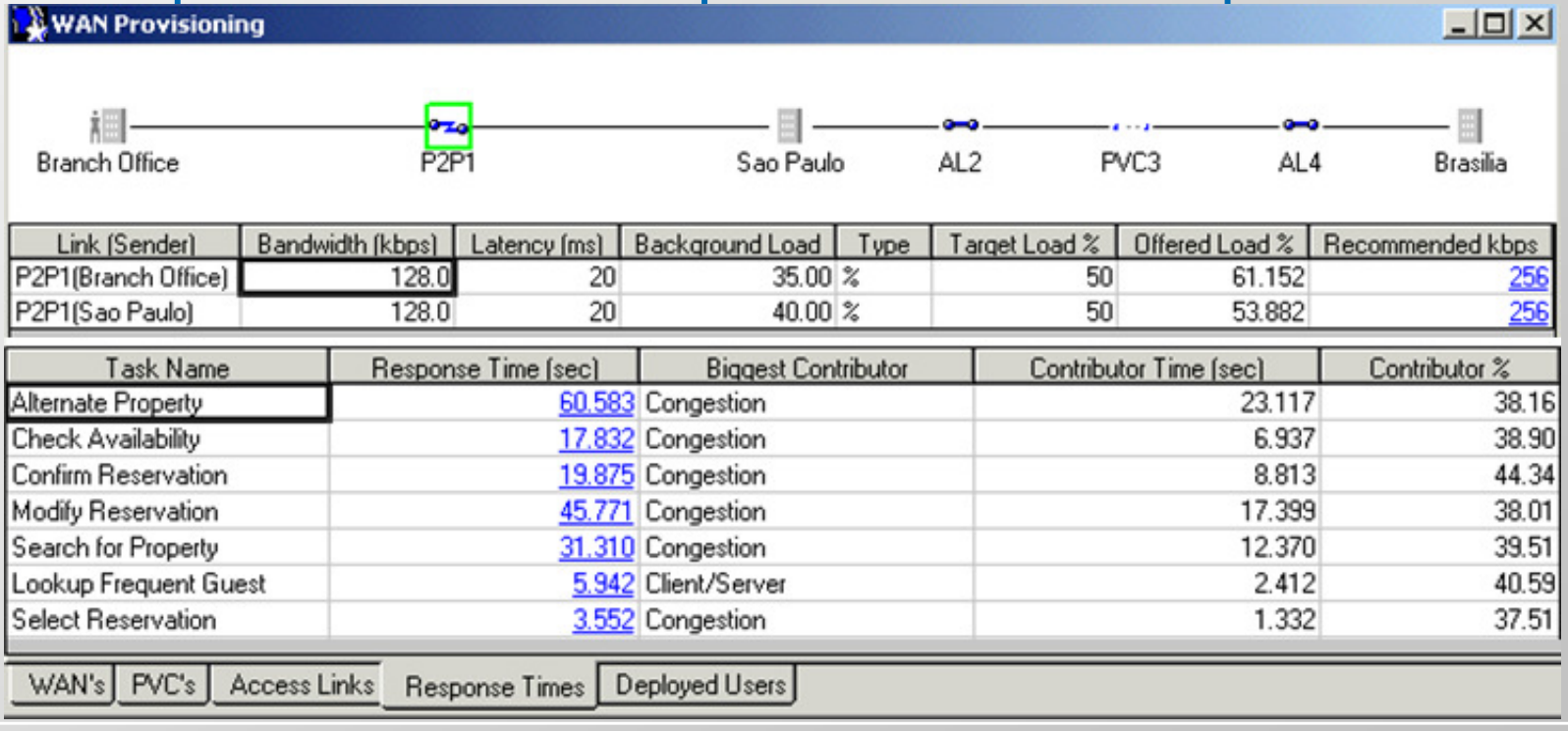
What to test in APA phase

Architecture verification

- Should we put this widget on the client or server
- Load balance, SSL offload, compression, caching …

WAN Provisioning

- User profiles x branch profile x network profile vs. response



Link (Sender)	Bandwidth (kbps)	Latency (ms)	Background Load	Type	Target Load %	Offered Load %	Recommended kbps
P2P1(Branch Office)	128.0	20	35.00	%	50	61.152	256
P2P1(Sao Paulo)	128.0	20	40.00	%	50	53.882	256

Task Name	Response Time (sec)	Biggest Contributor	Contributor Time (sec)	Contributor %
Alternate Property	60.583	Congestion	23.117	38.16
Check Availability	17.832	Congestion	6.937	38.90
Confirm Reservation	19.875	Congestion	8.813	44.34
Modify Reservation	45.771	Congestion	17.399	38.01
Search for Property	31.310	Congestion	12.370	39.51
Lookup Frequent Guest	5.942	Client/Server	2.412	40.59
Select Reservation	3.552	Congestion	1.332	37.51

WAN's | PVC's | Access Links | Response Times | Deployed Users

Performance Testing in APA phase

Load Testing

- Test the system against expected average & peak loads
- Can the system handle x transactions per hour?

Stress Testing

- Testing the system against unexpected loads (what if)
- Response under unexpected load, where & why it breaks

Scalability & Architecture Testing

- Aim for linear scaling, e.g. 100 more VUs = 1 more JVM & 3 JVMs per blade, memory per JVM…

Stability Testing

- Test over longer period – check for memory leaks and other stability issues

Outputs from APA against quality gates

Quality gates must be achieved before move to production

Outputs – Documented Performance Results (reports)

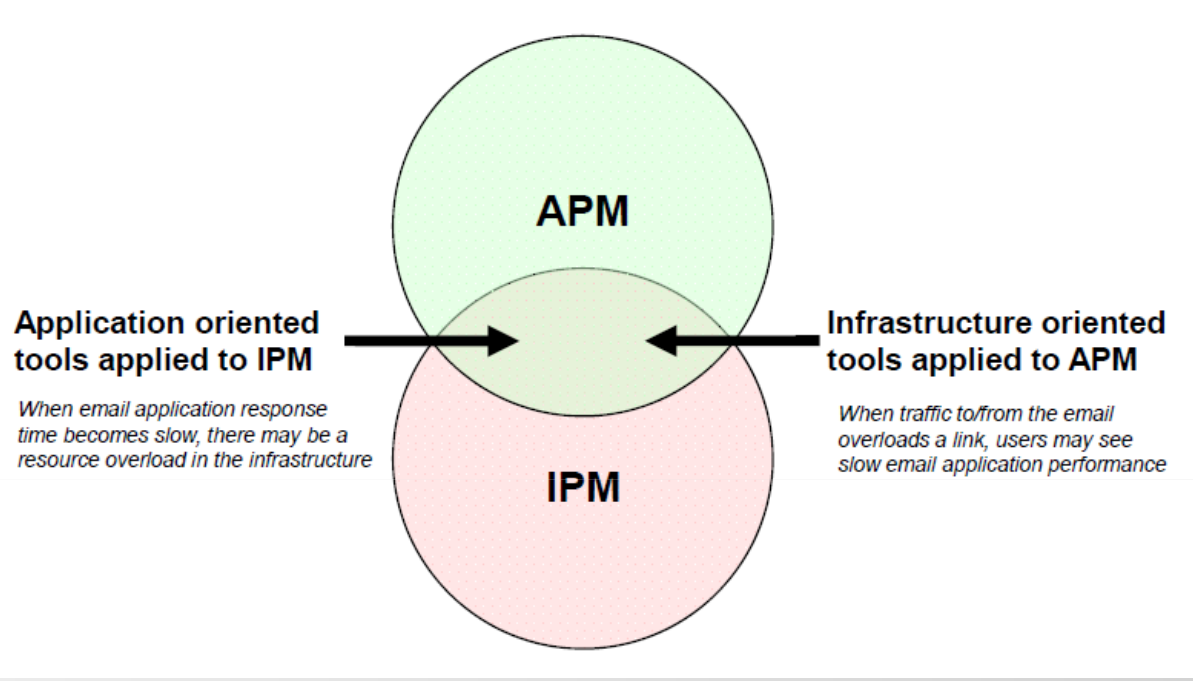
- All results must be compiled, documented and signed off
- Results must show conditions under which results were achieved including resource constraints, application settings, architecture…
- Report must show if requirements are met or not, including risk
- Must include exec summary showing the “bottom line”

Formal hand-over of QA results to production, information will serve as input to production monitoring

Ideally have a fact sheet showing the most pertinent criteria under test versus performance metrics met/not met

Outputs from APA phase become initial benchmarks for Application Performance Monitoring in Production

How to monitor in Production



Monitoring cause and effect

Relate service to infrastructure

How to monitor in Production

APM will tell you more about the service delivered

- More business centric, shows risk to business
- EUE measures user response, is all encompassing (end-to-end) & includes all dependencies. (whereas monitoring some dependencies may not mean you're aware of poor EUE)

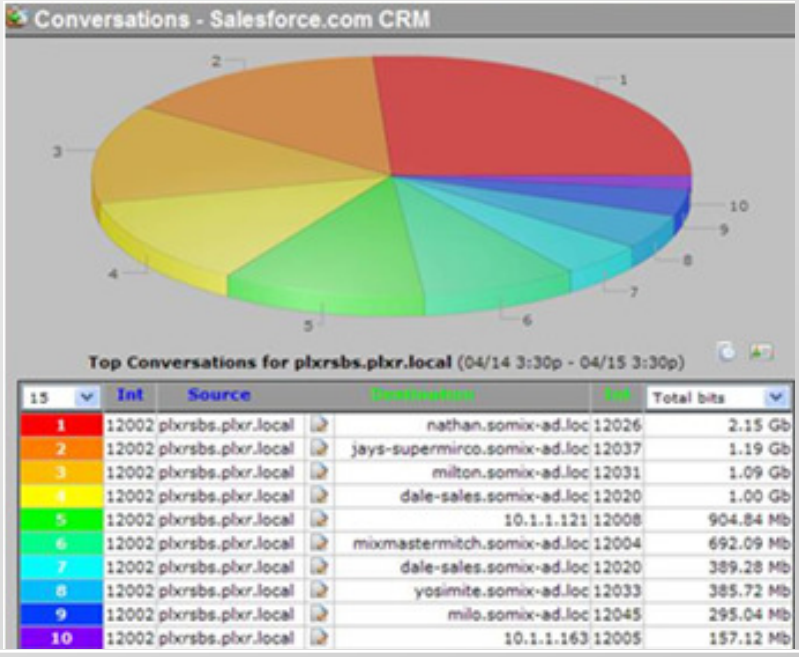
IPM will tell you more about the state of the infrastructure

- More technical, shows bottlenecks in resources
- By itself says very little about service delivered

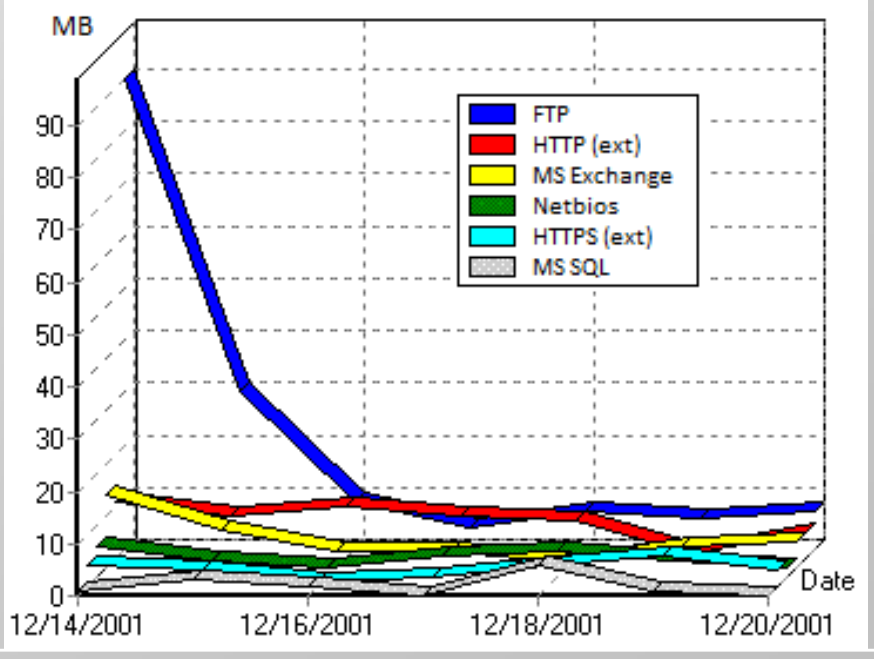
Application Performance Monitoring

Networked Application Traffic monitoring

- Take packet capture of transaction in production to confirm application is implemented as intended
- Traffic thresholds to ensure app uses BW as tested
- Network bandwidth & latency thresholds



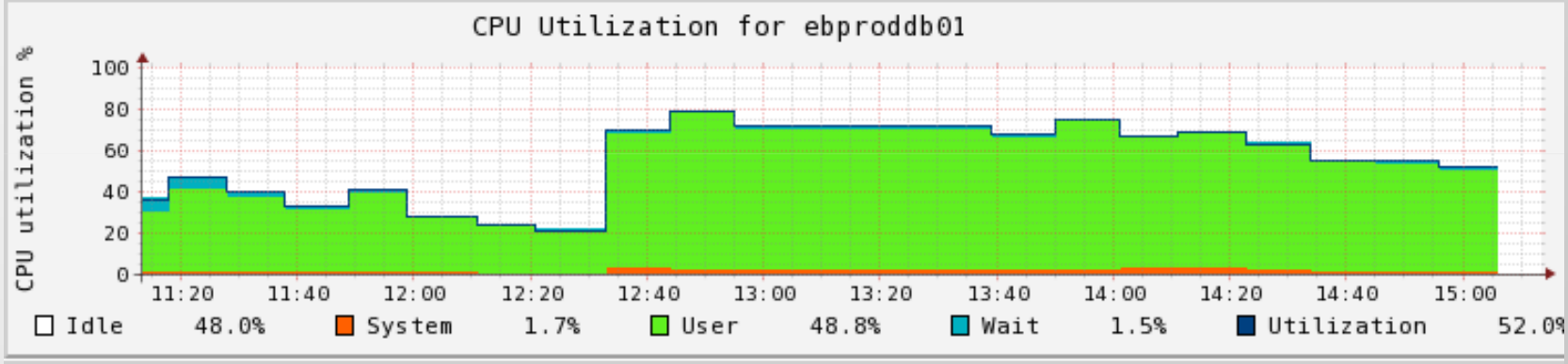
15	Int	Source		Destination	Int	Total bits
1	12002	plxrsbs.plxr.local		nathan.somix-ad.loc	12026	2.15 Gb
2	12002	plxrsbs.plxr.local		jays-supermirco.somix-ad.loc	12037	1.19 Gb
3	12002	plxrsbs.plxr.local		milton.somix-ad.loc	12031	1.09 Gb
4	12002	plxrsbs.plxr.local		dale-sales.somix-ad.loc	12020	1.00 Gb
5	12002	plxrsbs.plxr.local		10.1.1.121	12008	904.84 Mb
6	12002	plxrsbs.plxr.local		mixmastermitch.somix-ad.loc	12004	692.09 Mb
7	12002	plxrsbs.plxr.local		dale-sales.somix-ad.loc	12020	389.28 Mb
8	12002	plxrsbs.plxr.local		yosimite.somix-ad.loc	12033	385.72 Mb
9	12002	plxrsbs.plxr.local		milo.somix-ad.loc	12045	295.04 Mb
10	12002	plxrsbs.plxr.local		10.1.1.163	12005	157.12 Mb

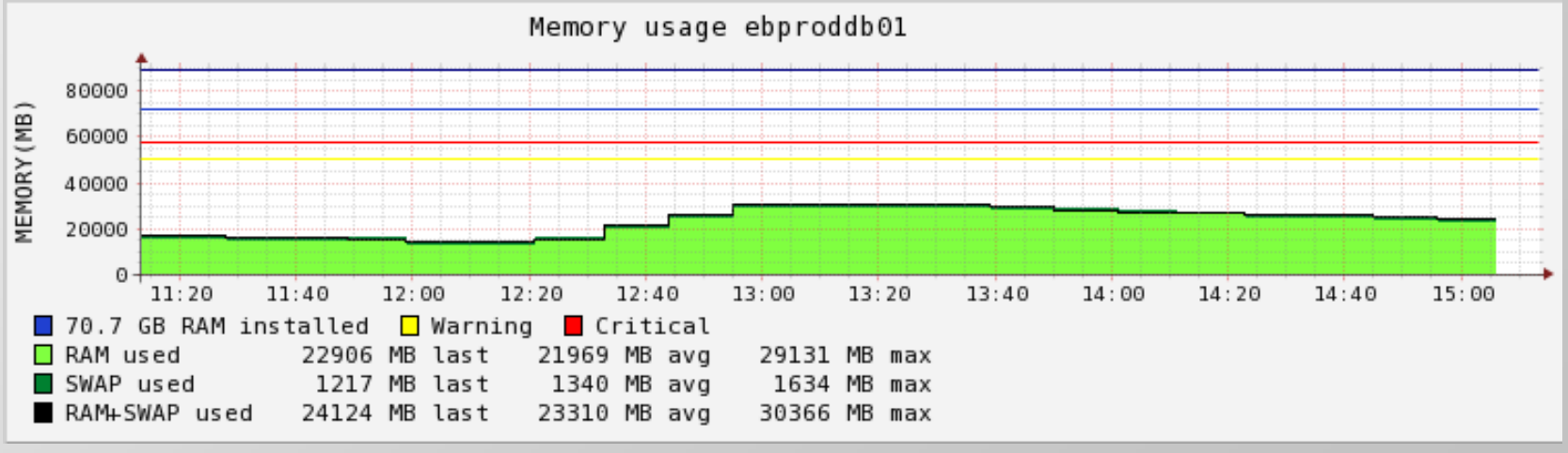


Application Performance Monitoring

Infrastructure Metrics

- Server/OS counters & thresholds as gleaned per testing
- Event logs & application logs

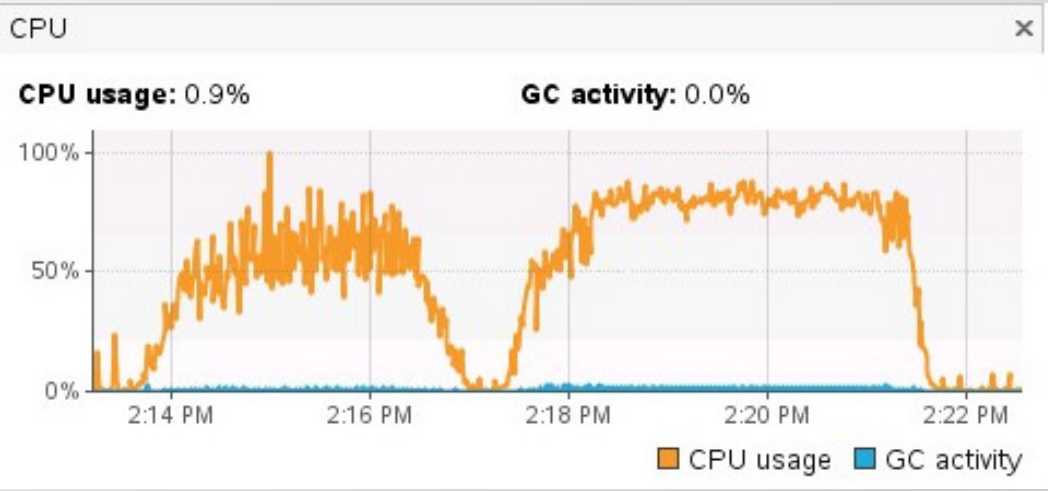


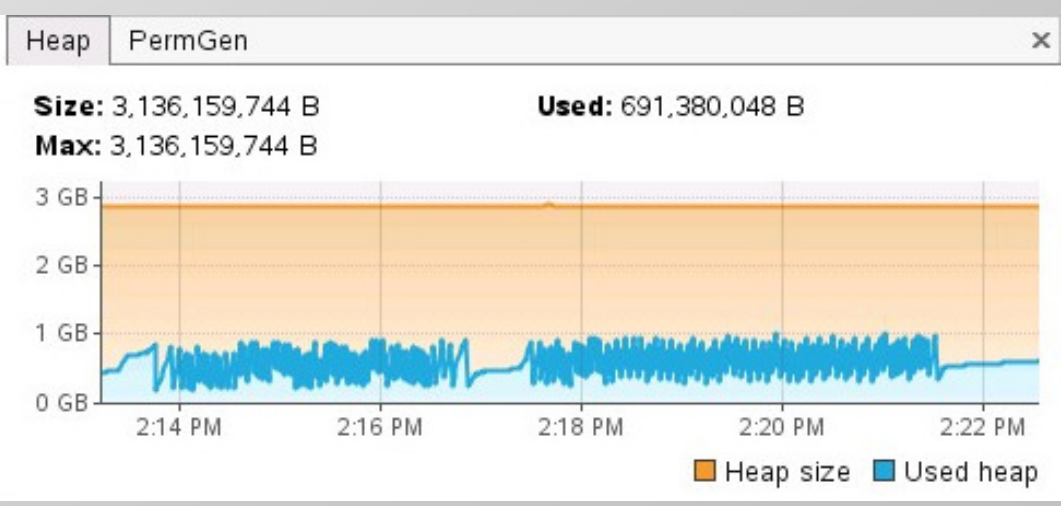


Application Performance Monitoring

Application Monitoring

- Application Specific counters and logs
- .NET/JAVA as required with tested metrics
- Profile CPU & Memory in heap over time and monitor resource intensive methods and backend calls

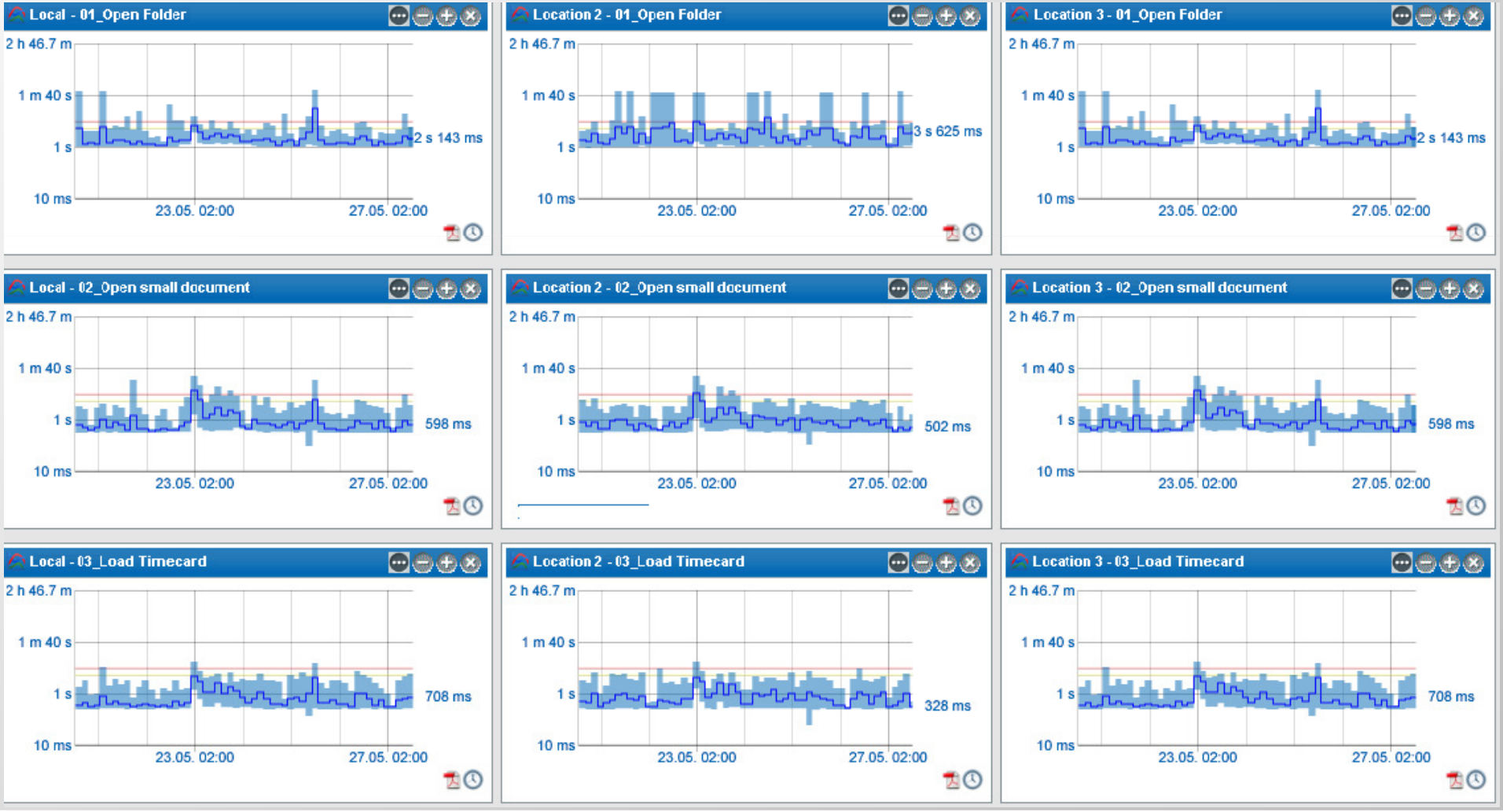




Application Performance Monitoring

End user experience monitoring against thresholds

- Trend all metrics as new (actual) benchmark & SLM



FDI & RCA - What's good & what's not

Performance issue – time is the only metric !

Set 3 types of thresholds (warning, alert, service level)

- Warning - When its been going worse than normal (benchmark – pro-active)
- Alerts – when its gone wrong
- Service Level - How much has it been going wrong (99.5%...should relate to service not infrastructure)

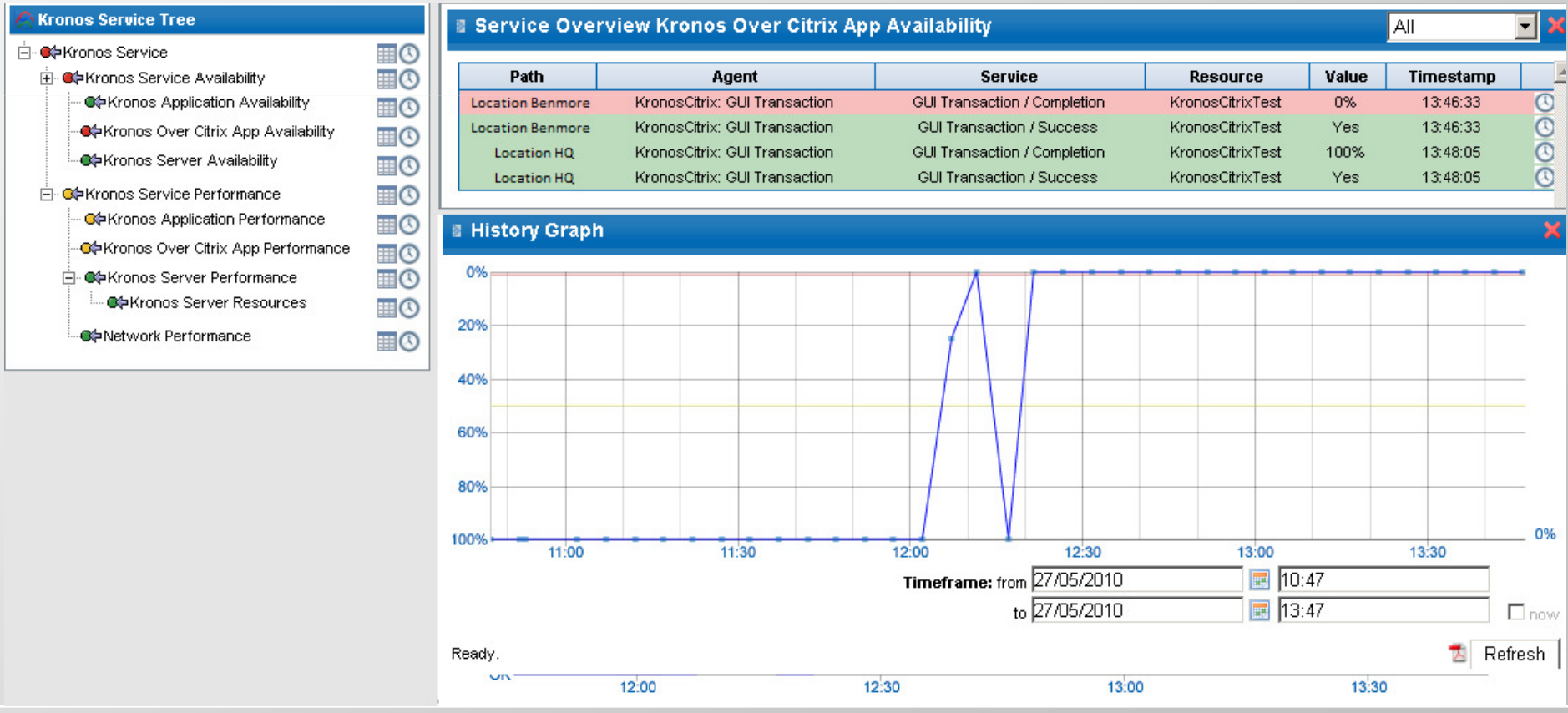
Categorise the performance problems

- Number of users affected, locations affected
- Number and type of business processes affected
- Risk to the business
- EUE has endless value here

Fault Domain Isolation

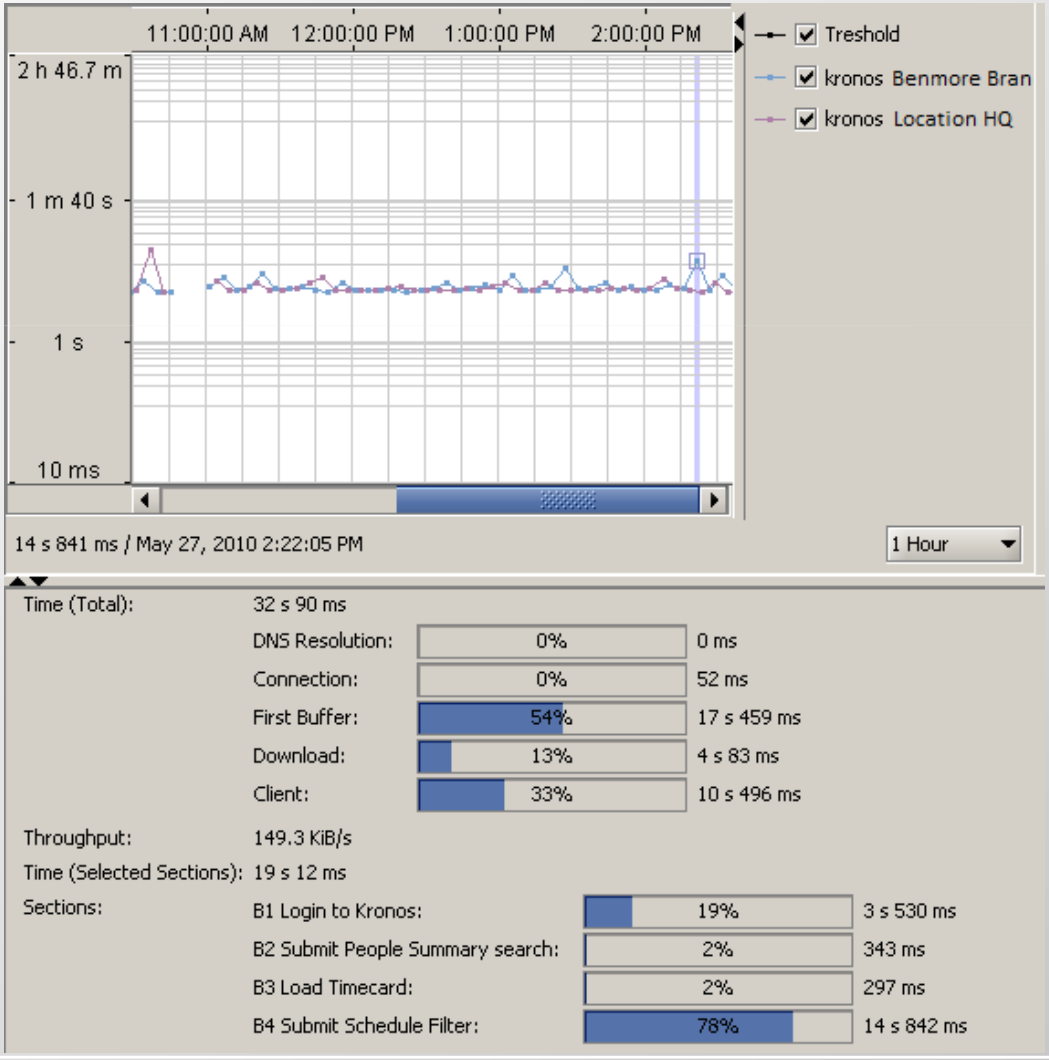
Classify problem to gain insight into impact & criticality

- Which service, location and how many users are affected? Availability or Performance
- Is the whole application not performing or a specific step



Fault Domain Isolation

- All IT departments are working in silos with network, server, desktop & application specialists



- Find the fault domain before finding the fault, if FDI is accurate, hand to correct silo
- EUE tools or manual multipoint packet captures are ways to establish fault domain

Root Cause Analysis

Network problem

- Check utilisation of affected link
- Check application breakdown of link if congested
- Check latency on link (if transaction is latency sensitive)

Server Farm problem

- Check utilisation of resources and causes
- Check for hung processes
- Consider DB locking

Application problem

- .NET/JAVA analysis to identify long running method
- SQL profiling to identify long running query
- Application logs & metrics

Outcomes from FDI/RCA

To do better, we need to learn from “mistakes”

- Document how problem was reported & classified
- Document how FD & RC were found
- Feed back Production trend & problem analysis to program management for review in future upgrades, architecture changes, application revisions…(Six Sigma or ITIL – continual service improvement approach)

APM and ITIL

The ITIL Information Technology Services Management (ITSM) framework defines processes

Most of the ITIL processes deal with:

- Implementation: delivering projects within schedule & budget
- Operations: keeping systems working
- Service model: Running IT as a service

Proper APM needs to determine the ITIL essential subset applicable to APM

- APM should not re-invent service process

Tools must be supportive of process and integrated as much as possible

Source: NetForecast, June 2010

ITIL Processes Related to APM

- **Service Operation**
  - **Event Management**
  - **Incident Management**
  - **Request Fulfillment**
  - **Problem Management**
  - **Access Management**
  - **Operational Activities of Processes …**
    - Change Management
    - Configuration Management
    - Release and Deployment Management
    - Capacity Management
    - Continuity Management
    - Availability Management
    - Knowledge Management
    - Financial Management for IT Services
- **Continual Service Improvement**
  - **Service Level Management**

Minimal set of processes that relate to ongoing APM delivery quality

Source: ITIL v3, May 2007

How APM supports ITIL processes

Incident: resolving performance degradation

- Trouble ticket system
- Actionable alarms with threshold controls
- Diagnostic capability

Availability: service operation

- Ability to define availability terms
- Historic data to determine trends
- Availability (service is working) reports
- Accessibility (users can access the service) reports

Source: NetForecast, June 2010

How APM supports ITIL processes

Capacity: proper service resources

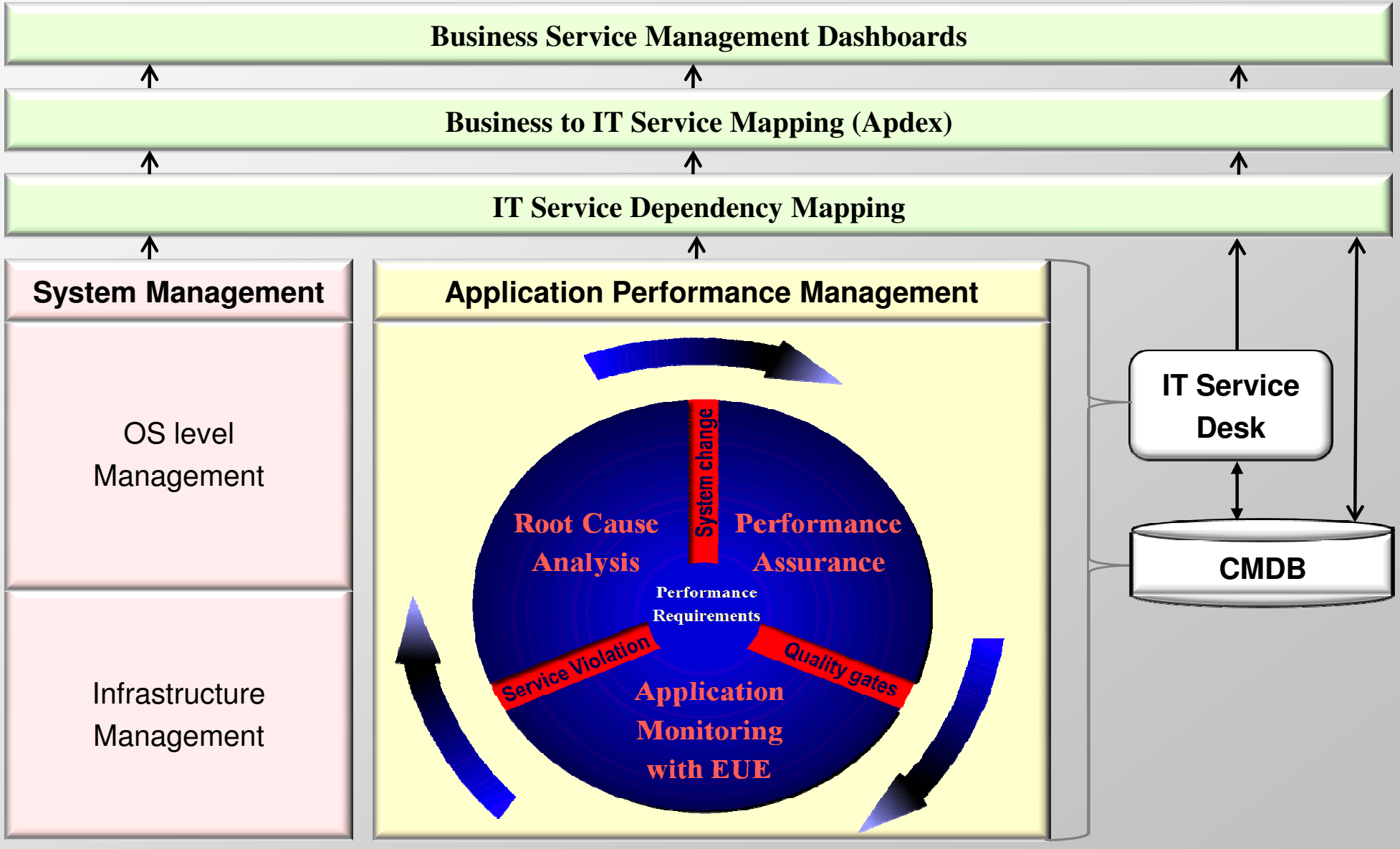
- Demand model (ability to project future demand)
- Demand –usage correlation
- Show over-utilized and under utilized resources
- Modeling “what-if” scenarios of change or growth

Service Assurance: meeting business needs

- Ability to define service targets – KPIs
- Continually monitor service and relate to targets
- Integration of incident, availability & capacity management data into single report
- Benchmarking & assessing the effect of change upon ability to meet targets

Source: NetForecast, June 2010

Where does APM fit into BSM



This “thingy” called APDEX

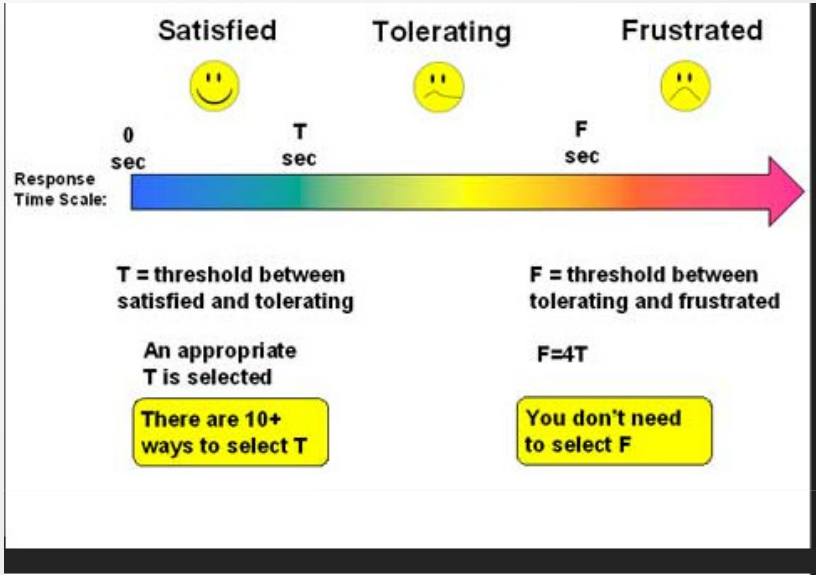
Apdex (Application Performance Index) is an open standard developed by an alliance of companies. It defines a standard method for reporting and comparing the performance of software applications in computing. Its purpose is to convert measurements into insights about user satisfaction, by specifying a uniform way to analyze and report on the degree to which measured performance meets user expectations.

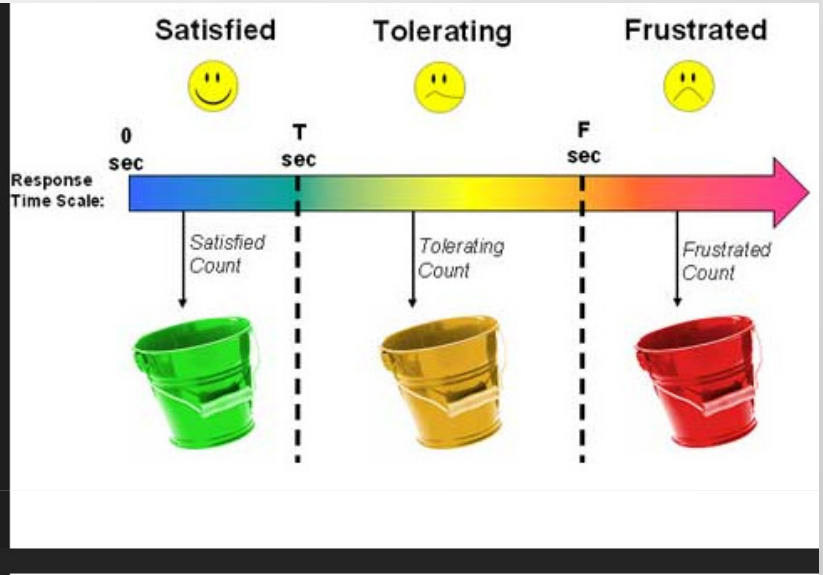
6.45	16.89	3.36	54.50	2.38
59.55	13.25	3.33	2.51	6.38
16.67	4.50	2.22	4.75	7.98
12.56	8.44	9.76	3.84	3.85
2.99	4.75	13.20	11.98	5.85
14.55	8.83	3.73	2.94	2.20
7.37	3.78	3.28	3.99	7.57
2.78	3.54	4.90	4.29	1.77
7.38	6.39	6.21	23.56	15.00
19.69	21.33	22.50	18.10	6.02
1.61	1.46	2.15	10.46	1.26
5.60	3.67	2.20	2.35	14.83
1.64	2.13	15.35	2.48	3.28
3.87	4.90	4.64	3.42	3.34
2.02	1.99	3.69	3.22	3.46
6.09	2.32	3.83	16.37	1.87
3.74	2.70	2.95	30.08	1.80
30.54	1.76	4.53	1.46	2.24
2.76	1.74	5.33	4.11	2.65
7.50	1.36	2.49	2.77	5.20

APDEX
Measuring What Matters™

Source: Apdex, June 2010

How APDEX works





Given
Target response time T and
Sufficient response time measurement samples
Then

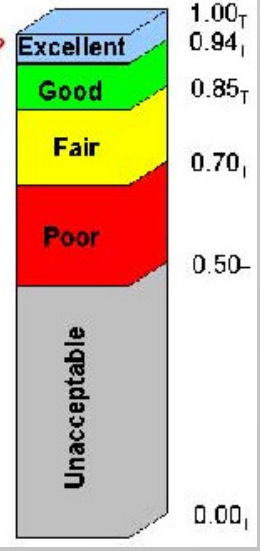
$$Apdex_T = \frac{Satisfied\ count + \frac{Tolerating\ count}{2}}{Total\ samples}$$

Note
Frustrated samples are not in numerator but are counted in total samples
Index
0 = Failure; 1 = Perfection (all users satisfied)

Now can you tell which application is in trouble?

	Apdex Scores				
	App A	App B	App C	App D	App E
Apdex	0.83_6	0.73_7	0.88_4	0.64_5	0.98_4

APDEX
Measuring What Matters™



Source: Apdex, June 2010

Questions & Answers