



BRENT OZAR
UNLIMITED®

Building a Health Check Report

When should you perform a health check?

As often as you want regular indicators of server health.

At least once a quarter.

- Regular indicator of system health.
- Spot hardware and software flaws before they stop the business cold.

Why run a health check?

You go to the doctor regularly:

- To make sure nothing is wrong.
- To get advice on your overall health.

Run a SQL Server health check for the same reasons!

It's an overview of system health.

It's your insurance policy

- Things can break at any minute.
- If you're running health checks and you know things are okay, a sudden failure really is sudden.

Why check your ISV applications?

Point the finger of blame in the right direction.

Break problems into two sections:

- Infrastructure problems
- Software problems

Promotes better communication with the vendor.

- Focus on making the product better.
- Do it before it's a problem.
- Stop complaining, start helping.

Think about your
last visit to the
doctor.

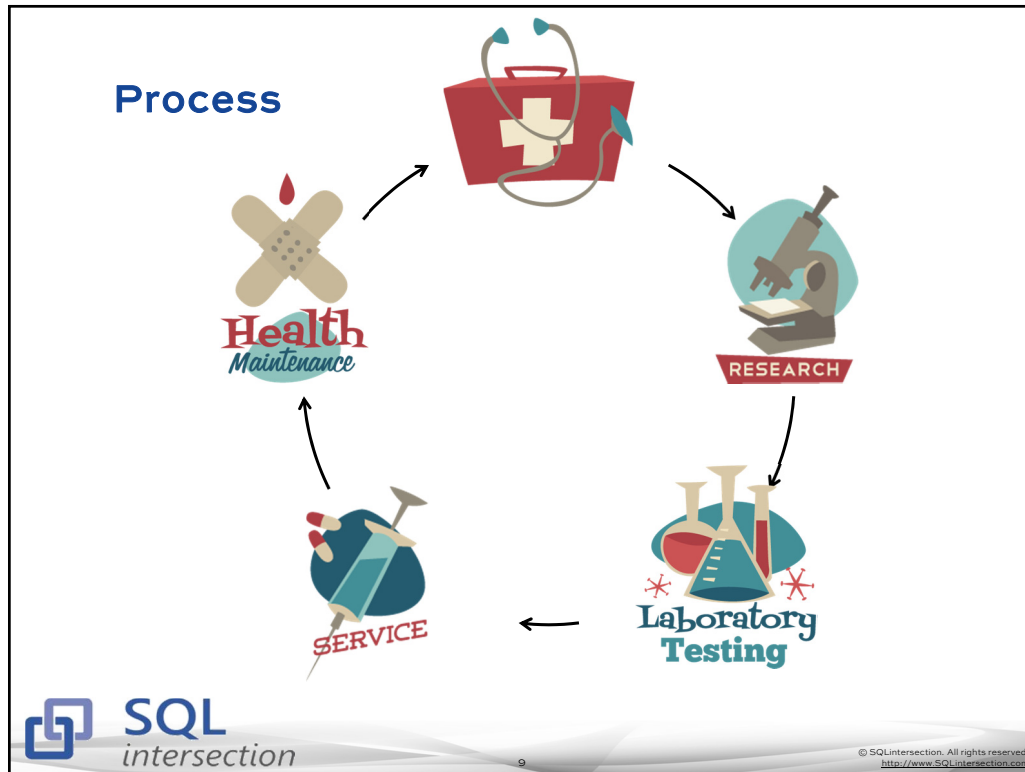
What kind of
process did they
follow?

A Visit to the Doctor

Does your doctor:

- A. Randomly poke you, look at various limbs/organs, ignore your symptoms, consult WebMD, and declare that you have bubonic plague and an obscure incurable cancer?
- B. Collect symptoms and data, perform research, recommend a course of treatment?

**SQL Server
is no different.**



What should your
health check
include?

SQL intersection

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

A Good Health Check Includes

1. List of symptoms
2. Evidence and vital signs
3. Details about recommended treatment
4. Prioritized action plan
5. A repeatable process

Symptoms



Collecting Symptoms

The goal of a health check is to fix users' problems.

Interview people to find out what's bothering them.

- Users
- Operations staff
- Business stakeholders

Get to the root of the problem quickly.

List of symptoms

Which application(s) are you using?

Which server(s) are involved?

Users – what are your current pain points?

Ops team – what are your current pain points?

Stakeholders – what are your current pain points?

If you want to get fancy, ask how this is preventing their success.

Symptoms: A Word of Warning

Symptoms are what the users perceive to be wrong.

- Don't confuse symptoms for the problem.
- A numb left arm doesn't always mean heart attack.

Symptoms help us prioritize recommendations.

Symptoms: A Word of Warning

Symptoms are what the users perceive to be wrong.

- Don't confuse symptoms for the problem.
- A numb left arm doesn't always mean heart attack.

Symptoms help us prioritize recommendations.

- ETL performance is incredibly slow.
- No users are affected.
- Should we recommend tuning ETL processes as a **high priority?**

Symptoms guide findings **delivery**, not discovery.

Symptoms & Delivery

Match your findings with users' concerns.

Symptoms of a problem are just that: symptoms.

- A symptom is subjective.
- Usually cannot be measured directly.

Correlate your findings with the symptoms.

- “You saw a large number of time outs in the application. We can correlate that with these long running queries and associated wait stats.”

Allocate Responsibility

With ISV applications, two groups are responsible:

- The vendor.
- You.

Allocate responsibility appropriately.

- Poor hardware cannot perform well.
- Fast hardware, however, can hide ugly code.

Gathering Vital Signs



Typical Vital Signs

1. Configuration

- Windows settings
- SQL Server settings
- Large, system-wide improvements can happen here.
- Verify configuration settings with tools like sp_Blitz®

Typical Vital Signs

1. Configuration

2. Index information

- Missing indexes can hurt performance.
- Extra indexes can also hurt performance.
- Collect index usage data

Typical Vital Signs

1. Configuration
2. Index information
3. Poorly performing queries
 - Identify the sources of performance problems.
 - Examine plan cache metrics first – it's passive and low impact to production.
 - A slow report may be caused by frequent OLTP queries.

Typical Vital Signs

1. Configuration
2. Index information
3. Poorly performing queries
4. Top wait statistics
 - Waits provide insight into why queries are slow.
 - Identify top waits over time and by average wait length.
 - Some waits may only occur during activity – sample regularly.

Typical Vital Signs

1. Configuration
2. Index information
3. Poorly performing queries
4. Top wait statistics
5. Storage metrics
 - Find signs of storage contention.
 - Compare throughput to latency.

Typical Vital Signs

1. Configuration
2. Index information
3. Poorly performing queries
4. Top wait statistics
5. Storage metrics
6. Memory utilization
 - Is there enough memory to cache data?
 - Is memory being used by the plan cache?

Typical Vital Signs

1. Configuration*
2. Index information*
3. Poorly performing queries*
4. Top wait statistics
5. Storage metrics
6. Memory utilization
7. All kinds of other stuff

A * means we'll cover it today

Our Tools

Configuration – sp_Blitz®

<http://brentozar.com/blitz/>

Indexing – sp_BlitzIndex®

<http://brentozar.com/blitzindex/>

Poorly performing queries

<http://brentozar.com/go/plancache>

Let's grab vital signs!

Analyzing the Data



What story are we going to tell?

Understand success

If it's not related to pain, don't spend time on it.

- You don't have to document that 1AM ETL is slow if people are worried about mid-day inventory queries.
- You do have to document supporting evidence.

If it is related to success, make sure you include it.

- No backups, no DBCC checks – let someone know.

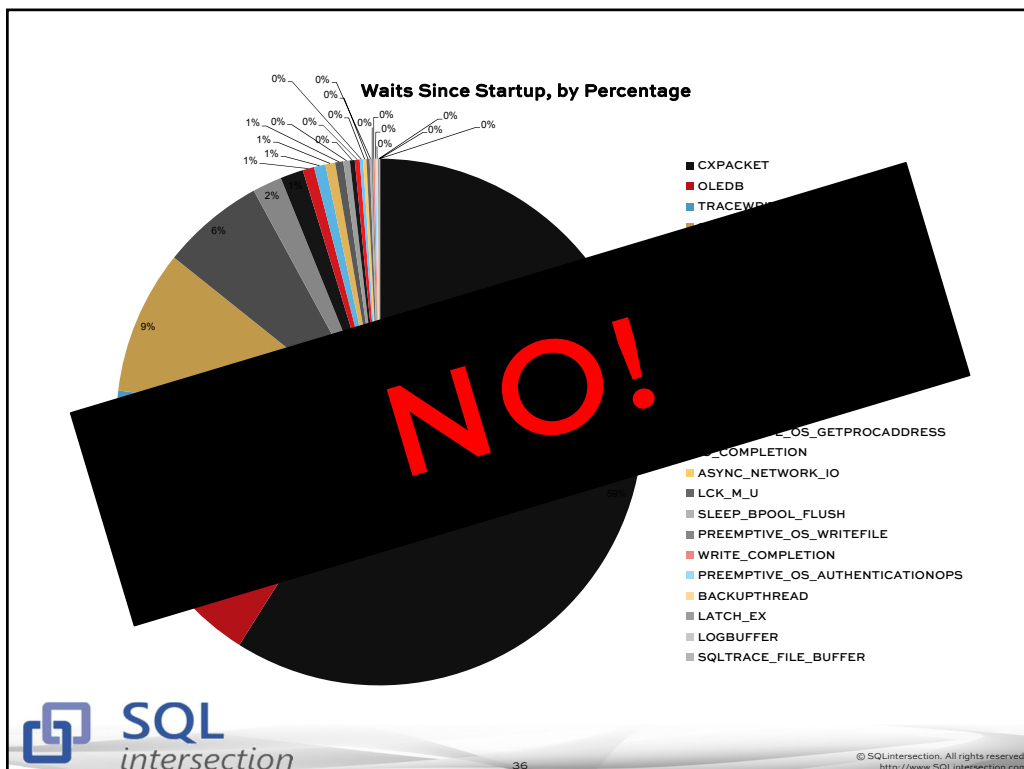
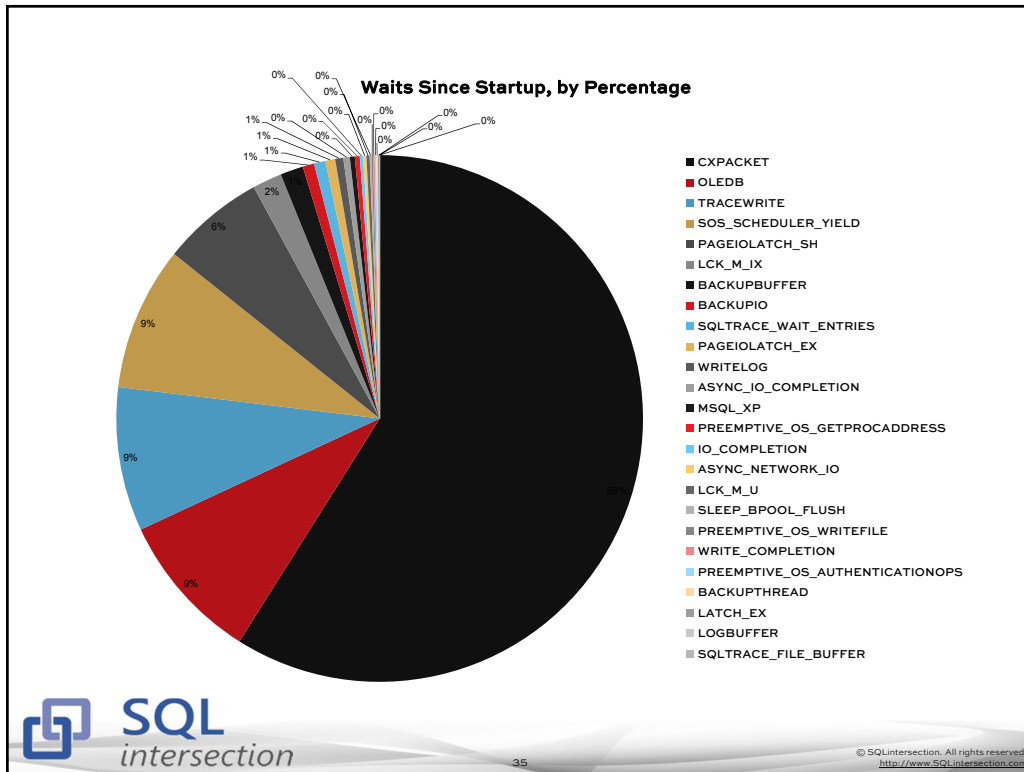
Correlate findings for long term success.

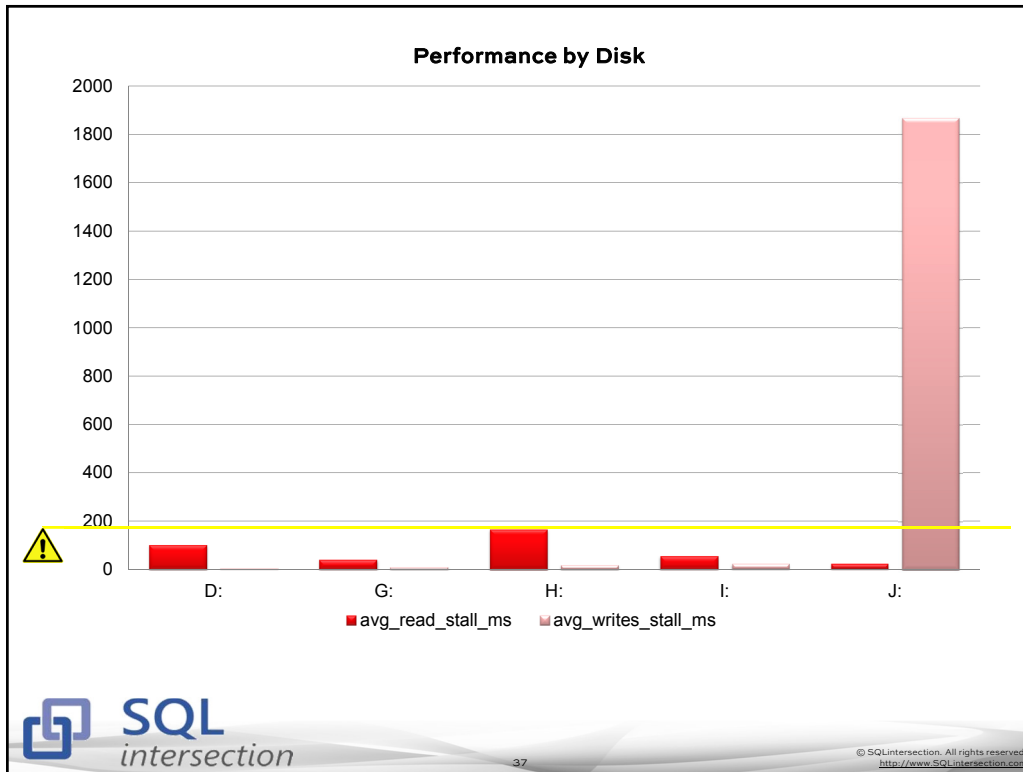
What should we show key stakeholders?

Focusing the Message

Narrow down your message to a few salient points.

Deliver the message based on evidence.





What do we learn?

Convey a clear message to business users & the ISV.

Use simple, understandable images to prove your points.

Isolate problems into clear responsibility areas:

- “Boss: We need more/better storage so reports will be faster and they don’t block business users.”
- “Vendor: We’d appreciate you supporting RCSI, readable secondaries, or a separate reporting schema.”

Solving the Problem



Detail Solutions

Create a list of possible solutions

- Match the solution to a symptom and pain.
- Detail a list of risks with the solution.
- Mitigate each risk.

People like options

- Even when the options are “get better or stay sick”.
- Provide a local option for every vendor fix.
Sometimes we can fix the vendor’s code problem with faster cores or more RAM.

“Stop doing that!”

Don’t just point, provide a road map.

Proposing solutions builds trust.

Describe anti-patterns that can get in the way.

- Is the ISV writing cursors?
- Is there no index maintenance?
- Is there too much index maintenance?

Make a map to **success**.
Point out failure pitfalls.

Create an action plan

Provide a solution that can be accomplished in stages.

- Some things will be clean up.
- Others will be small projects.
- There may even be large projects.

A prioritized plan makes implementation easier.

- Short, medium, long term.
- Time frame for each stage.

What's the vendor's plan?

Detail the problems you found.

- Be specific.
- Describe problem queries and provide steps to reproduce.

Be aware that your priority may not reflect the vendor's:

- Big fixes won't be fast.
- What seems trivial may be hard.

Do push for a roadmap, but know that specific dates may be hard to get.

No surprises

Everything in the action plan must be documented.

- Provide supporting evidence.
- If it doesn't relate to the pain points, throw it out.
- If it doesn't relate to success, throw it out.

Solutions should be reasonable.

- The solution shouldn't be "buy new servers and SAN" or "write a new app".
- Focus on measurable goals.

Meticulous Documentation?

Someone else already wrote an overview.

Provide a recommended reading/viewing list.

- Don't reinvent the wheel.
- Don't re-explain virtual log files.
- **Do** provide helpful, relevant links.

Highlight problems and point people to another explanation when you can't summarize quickly.

Staying in Shape



Success Criteria

How do you know you're getting better?

Define success criteria

Recommendations should include success metrics.

Examples:

- Reduction in PAGEIOLATCH_*
- I/O latency should be reduced
- Users will stop complaining

Problem Detection

Perform regular server reviews.

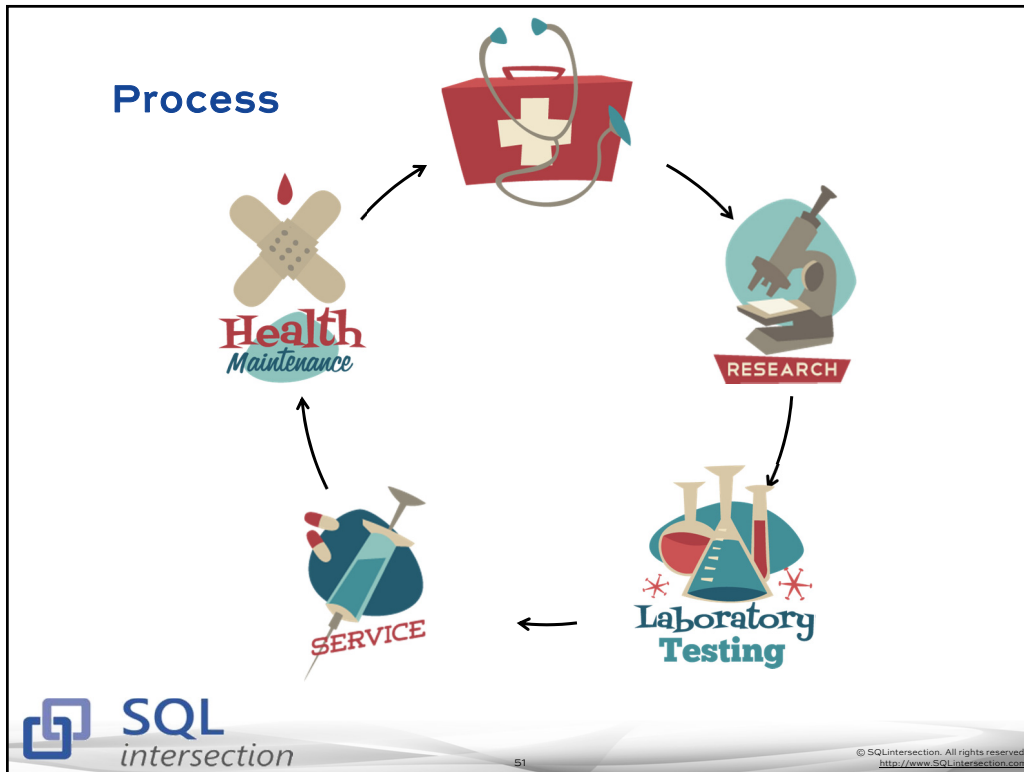
Use high level scripts to find major problems.

- sp_Blitz®
- sp_BlitzIndex®

Only dig deeper as needed.

- Look for problem indicators.
- When you find them, perform a deeper analysis.

SQL Server health is a cycle.



Ensure Database Health

1. Perform regular health checks
2. List of symptoms
3. Capture evidence and vital signs
4. Document recommended treatment
5. Create a prioritized action plan

SQL intersection

52

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Your vendor is a partner

Create a separate set of changes for them.

Share your own fixes.

- Your changes may help them improve other deployments.
- Plus it shows you're working with them.

Where possible, provide a local mitigation to ISV issues.

Free Tools

<http://brentozar.com/blitz/>

<http://brentozar.com/blitzindex/>

<http://brentozar.com/go/plancache/>



BRENT OZAR
UNLIMITED®

ExampleCo SQL Critical Care®

What Does Success Look Like To You?

Improved concurrency.

Ability to handle more users and spikes in load.

Long term stability.

Your Environment

EC2 instance

- 4 vCPUs
- 34GB memory

Modern software

- Windows Server 2012R2
- SQL Server 2014

General Health Issues

Some high level health items have not been set up:

- DBCC checks
- Corruption alerts
- High severity alerts

Mitigation:

- Use sp_Blitz® output or our setup guide to correct these configuration issues.
- See <http://brentozar.com/blitz> or <http://brentozar.com/go/setup> for more.

You aren't running backups!

You've never backed up any of these databases.

Run, don't walk, to <http://ola.hallengrent.com> and set up automated backups right now.

What are we waiting on?

Massive Parallelism

Many queries are using up a lot of CPU.

This is inhibiting concurrent querying.

The solution is to tune parallelism through:

- Effective indexes.
- Query tuning.
- Controlling parallelism with server level settings.

More CPU Contention

There are a lot of SOS_SCHEDULER_YIELD waits.

In effect, queries are fighting to get on CPUs.

Mitigations:

- Query re-writes to avoid common anti-patterns.
- Maybe some indexing – faster queries will use fewer resources.

Signs that disks could be a problem

They're not the top problem now.

But read and write stalls are higher than expected.

Mitigations:

- Use the right indexes.
- Tune queries to avoid implicit conversions.
- Consider switching to SSDs
(this is the new default in AWS).

Wait Stats

Wait Type	% Total Wait Time	Number Waiting Tasks	Avg Wait Time (ms)
CXPACKET	79.6	2,003,890.0	4.10
SOS_SCHEDULER_YIELD	12.3	799,842.0	1.60
PAGEIOLATCH_SH	5.0	20,580.0	25.30
IO_COMPLETION	1.2	21,812.0	5.60

Query Problems

sp_BlitzCache™ found some problems

Implicit conversions

- SQL Server is converting values on the fly.
- This can lead to poor index use.

Parameter sniffing

- Big variance in query performance, this can be caused by data skew.

Missing Indexes

- Top slow queries have missing index requests.

Queries to tune

TopPeopleForTag

- 3 missing indexes
- Parameter sniffing
- Implicit conversions

CommentsByKeyword

- Parameter sniffing

Action Plan: Us

Short Term: 1 week

Start running DBCC checks.

Start running backups.

Implement high value missing indexes.

Fix basic setup

- <http://brentozar.com/go/setup>

Medium Term: 1 month

After tuning indexes, examine wait stats.

- Tune for parallelism. If CXPACKET persists:
 - Increase cost threshold for parallelism.
 - Set MAXDOP appropriately (start with 2).

Long Term: 1 Quarter

Evaluate disk latency:

- Consider optimizations provided by Amazon.
- Provisioned IOPS.
- EBS SSD volumes.

Action Items: VendCo

Vendor Action Items

Tune worst queries:

- TopPeopleForTag
- CommentsByKeyword

Evaluate a separate reporting system.

- Consider a data warehouse system.
- Consider supporting scale out reads with AlwaysOn AGs

Consider adding the missing indexes outlined in the short term section.