Runtime Analytics

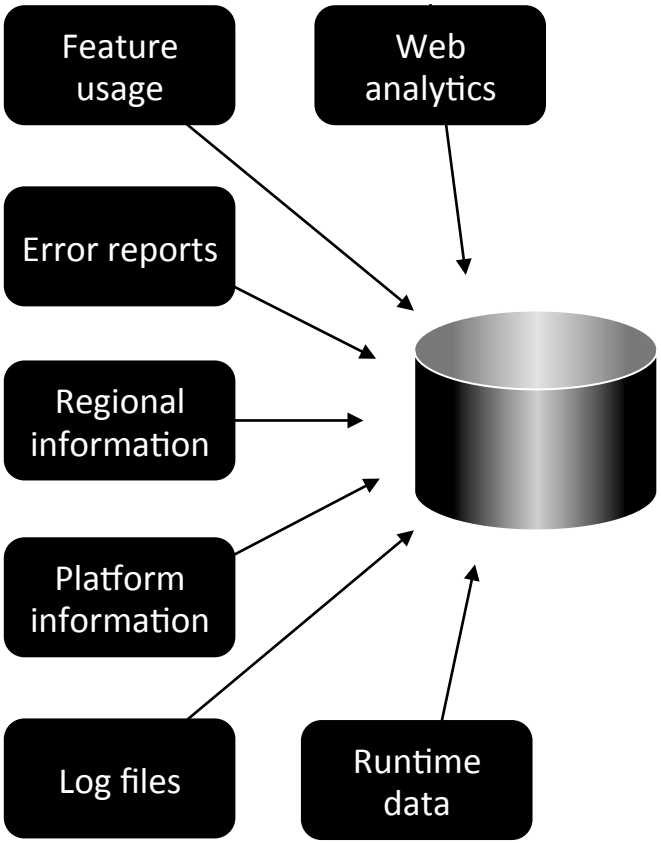
or finding out what your users really think of your software

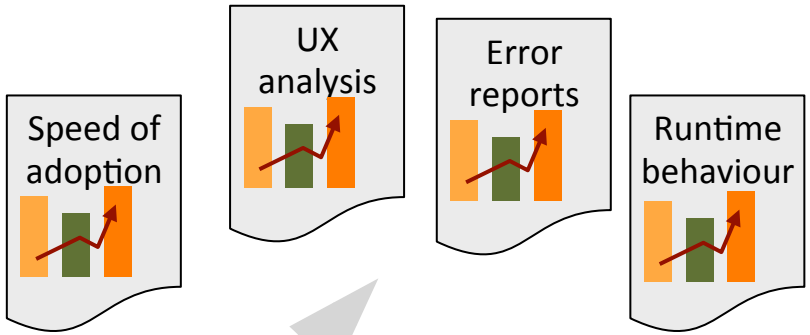
Jonathan Allin
Product manager and evangelist

redgate®
ingeniously simple tools

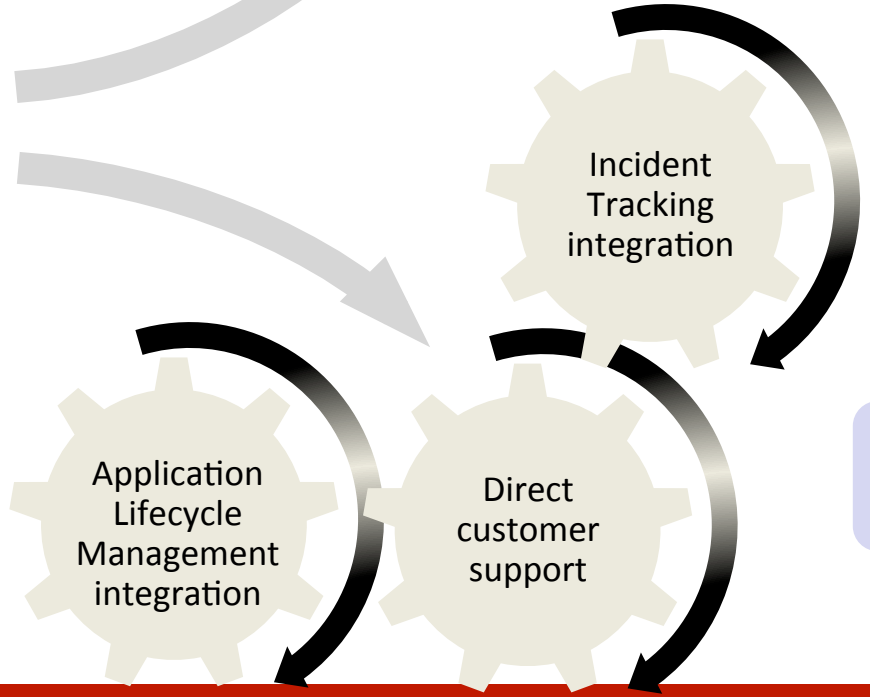
A definition for Runtime Analytics

1. Collects and stores comprehensive information about how users interact with your applications

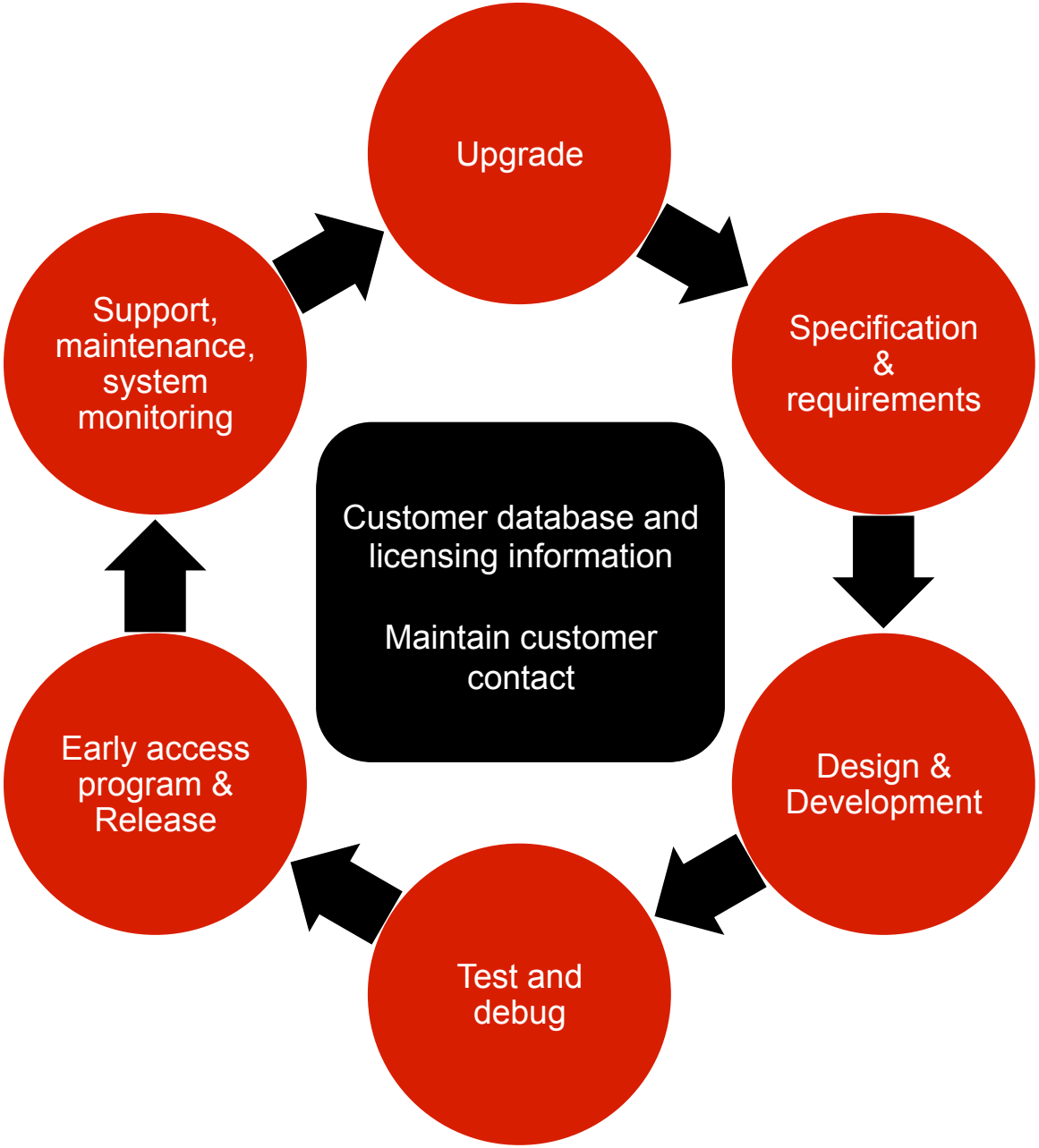




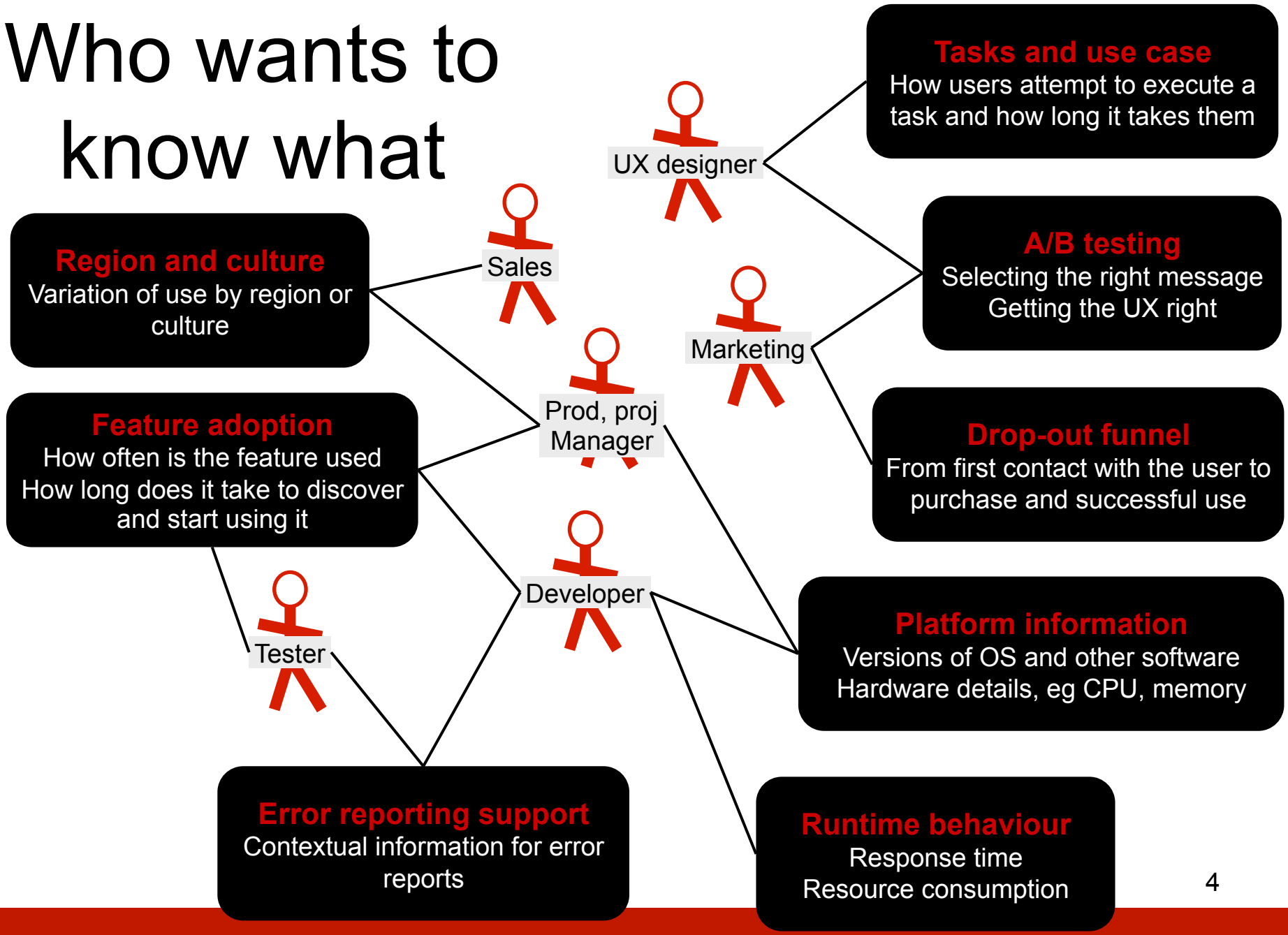
2. Provides functions and reports to analyse the information



3. And tools that act on the information

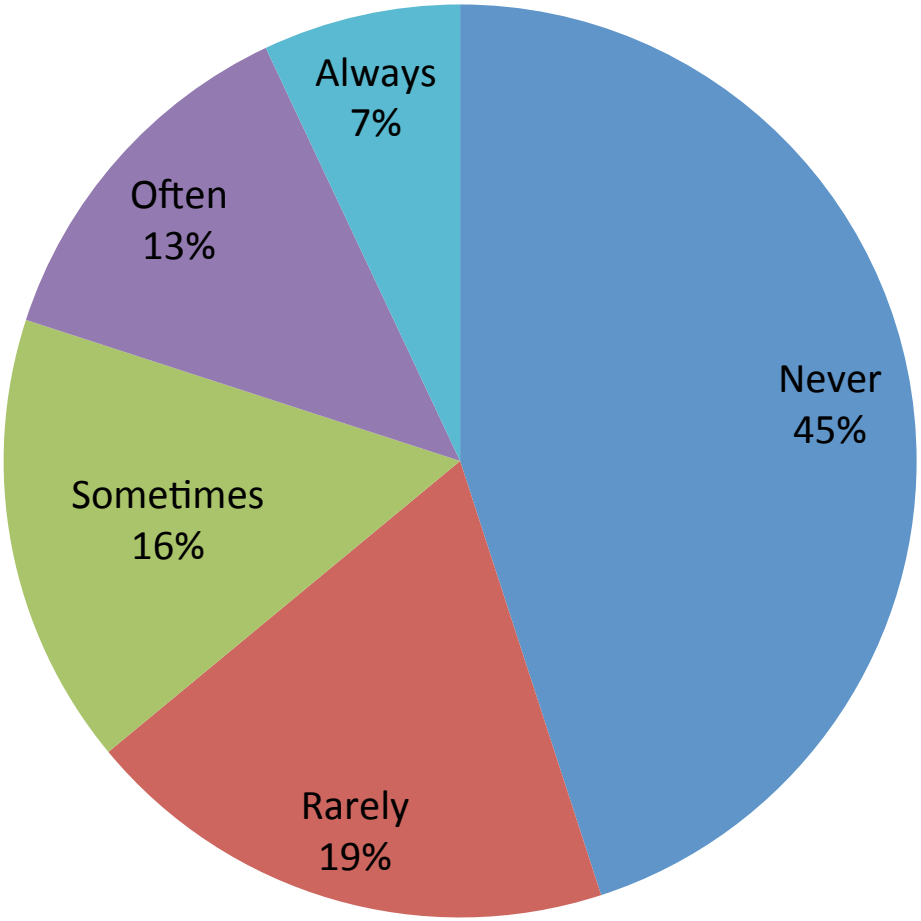

Upgrade
Specification & requirements
Design & Development
Test and debug
Early access program & Release
Support, maintenance, system monitoring
Customer database and licensing information
Maintain customer contact

Who wants to know what



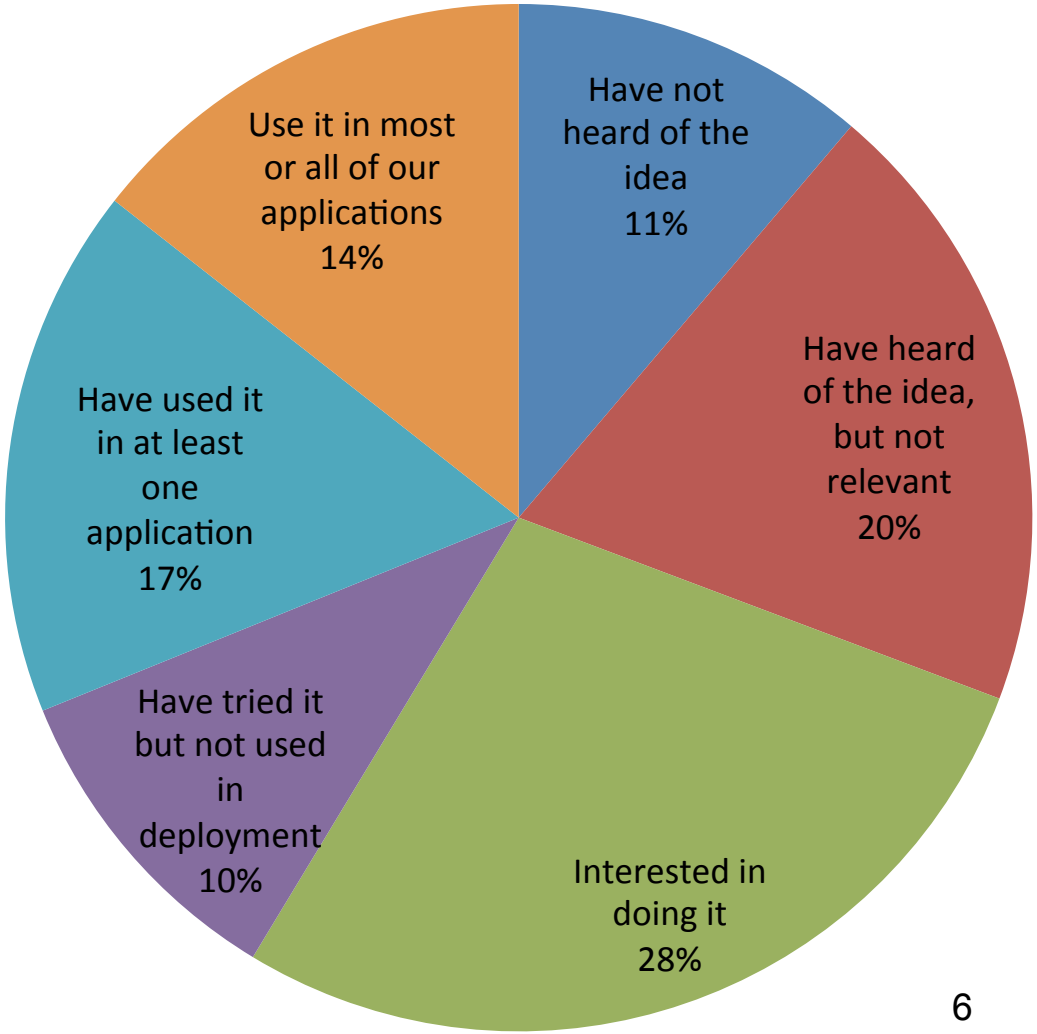
We need to know what's valuable

- How often is a feature used?
  - *Data from The Standish Group*



Do you ever automatically gather data on how your applications are used?

- Only 14% of respondents make regular use of runtime analytics
  - *Data from Red Gate survey. ~220 respondents*



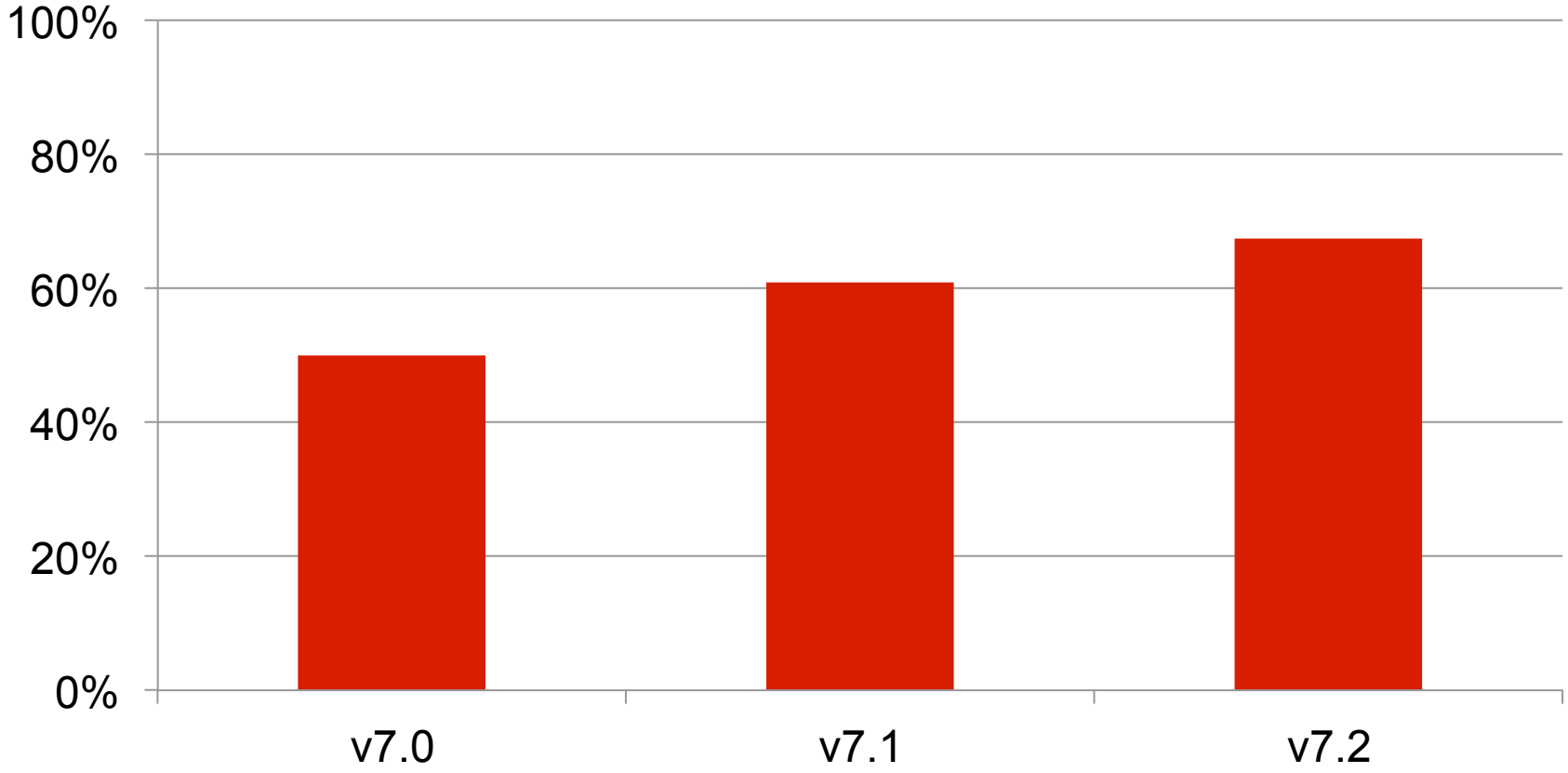
Two contentions

1. We should all be concerned about how our customers use our products
   - and how our products behave in the field

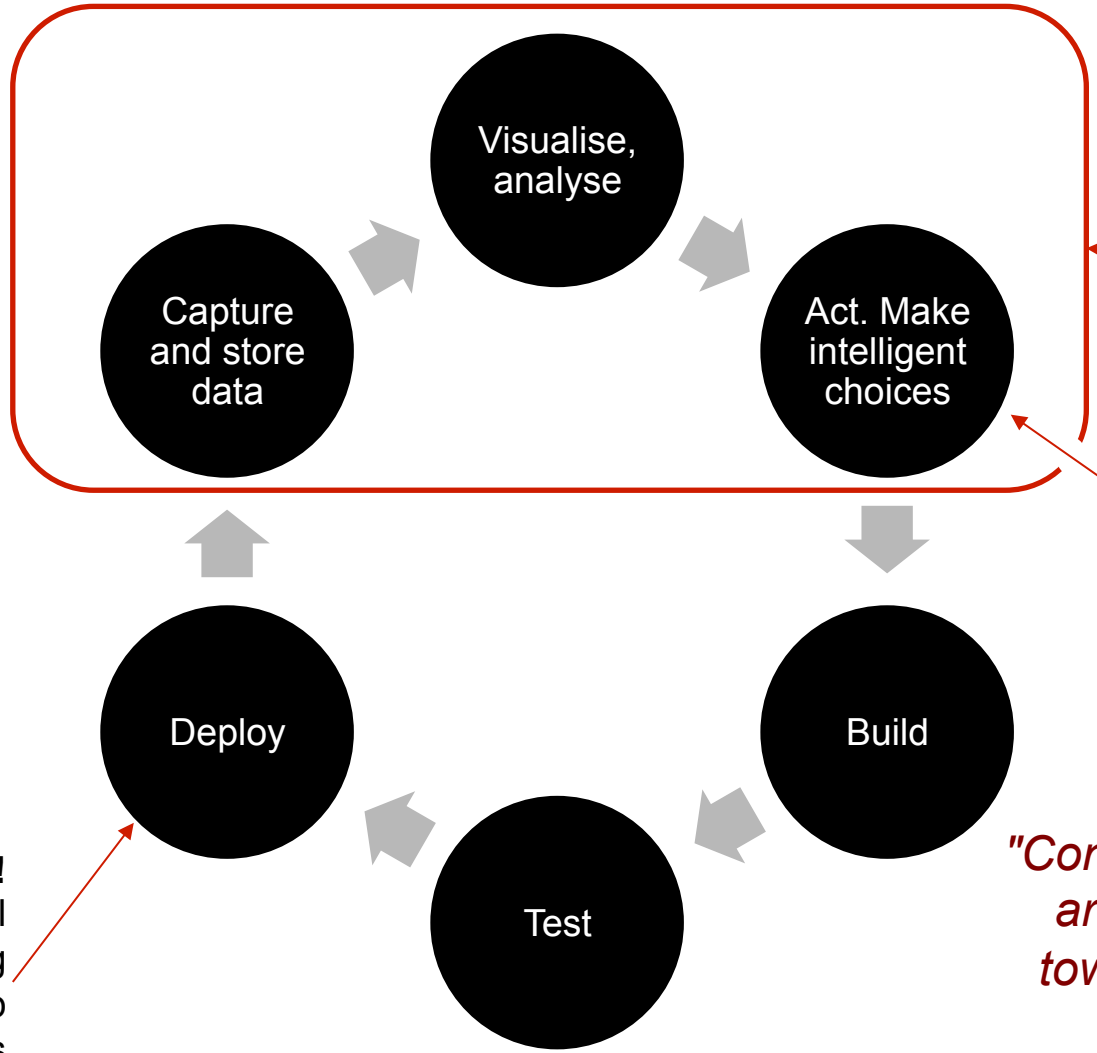
2. Runtime analytics will multiply ROI
   - It's not just an incremental benefit

ANTS Memory Profiler

Percentage of sessions in which a memory snapshot was taken by new users



Build Measure Learn



Potential road bump!
Install
CfU, licensing
Help
Web pages
Shopping cart

"Continuous analytics are an early step toward continuous feedback"
(Forrester)

Lean Startup

- You have a promising idea
- But there are lots of unknowns
  1. Is there a market
  2. Can you reach the market
  3. Is it technically feasible
  4. ...

Lean Startup in principle

- Get answers to the big unknowns as quickly and painlessly as possible
- Break down these unknowns into experiments of one or two weeks
  - Form a testable question
  - Run the experiment
  - Adapt the product

The "product" could be a marketing message, a report mock-up, a magic website, or a minimal implementation

An "experiment" can be an investigation, updating a website, releasing the next version, ...

Lean Startup in practice

- Experiments replace Agile Sprints
- Fast validated learning cycles

Build ⇨ Measure ⇨ Learn

- Swarming
  - Can appear inefficient
  - But maintains velocity
  - Reduces work in progress
- Fix as you go
  - Don't stack up issues and faults in a tracking system
- Weekly discussions
  - Review results of completed experiment(s)
  - Adjust expectations
  - Build new experiments to test new expectations

Runtime Analytics

Provides automatic data collection (from website or application)
Keeps your key metrics up-to-date and visible to the whole team
Augments, but doesn't replace, direct customer contact

Unknown #1: Is there a market?

- Can you identify a value proposition for your potential market?
- Magic website
  - Minimal content
  - Find out if customers are interested
    - Measuring hits provides little information
  - Collect email addresses
    - Don't ask for lots of details
- Talk to a few people who represent the market
  - ie are proxies for the market
  - Present the simplest possible product
    - eg mocked up screen shots, Flash demo, paper reports, a verbal picture, ...
  - Is your proposition of value to them?
  - If not, modify your proposition (or bail out)

Building the simplest product

- MVP: Minimal viable product
  - Has a value proposition
    - ie customers would pay something for it
  - Can be implemented quickly and easily
    - May deliver just a single feature
    - No bells and whistles
- Measure the response
  - Runtime analytics
  - Talk to your market proxies
- Learn, eg
  - Is it being used as you expected
  - Is it being used successfully
  - Is it sticky
- Continue around the loop
  - Modify and extend the product based on what you have learnt
  - Implement these changes as MVPs

Don't worry about technical debt

Unknown #2: Can you reach the market?

- How do you get your potential customers to notice you or take you seriously?
- Do you have to go through a reseller or other intermediary?
  - What are their pain points?
- Use your market proxies
  - Find out which journals they read, which conferences they attend, which websites they use, what they search for, their user groups, who they listen to, how they buy, ...

Unknown #3: Is it feasible?

- Do you have, or can you acquire, the necessary resources, knowledge, processes, to build your value proposition?
- Test the high risk areas - in the field
  - Measure runtime behaviour
- Don't attempt to build a complete solution
  - Use prototypes, experiments, etc

Measure what matters

Avoid "vanity" metrics

What matters

- Customer benefit
- Retention (or stickiness)
- Speed of adoption
  - Of a feature or application
- Success rate for completing a task
  - And how long it takes

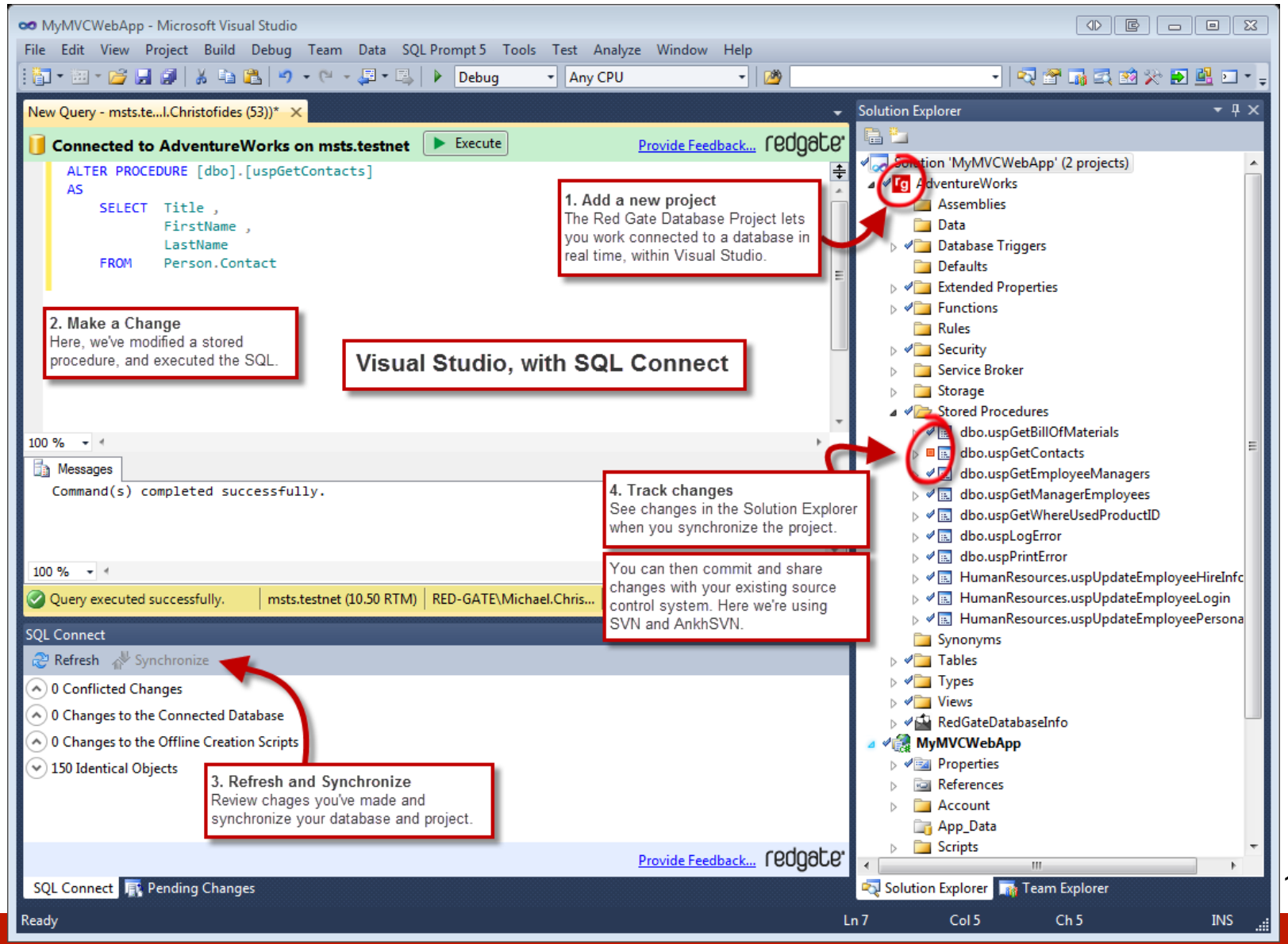
What probably doesn't

- Metrics that don't teach you anything
- Or that can be the result of a campaign
- Examples
  - Unique visitors
  - Number of downloads
  - Sales growth

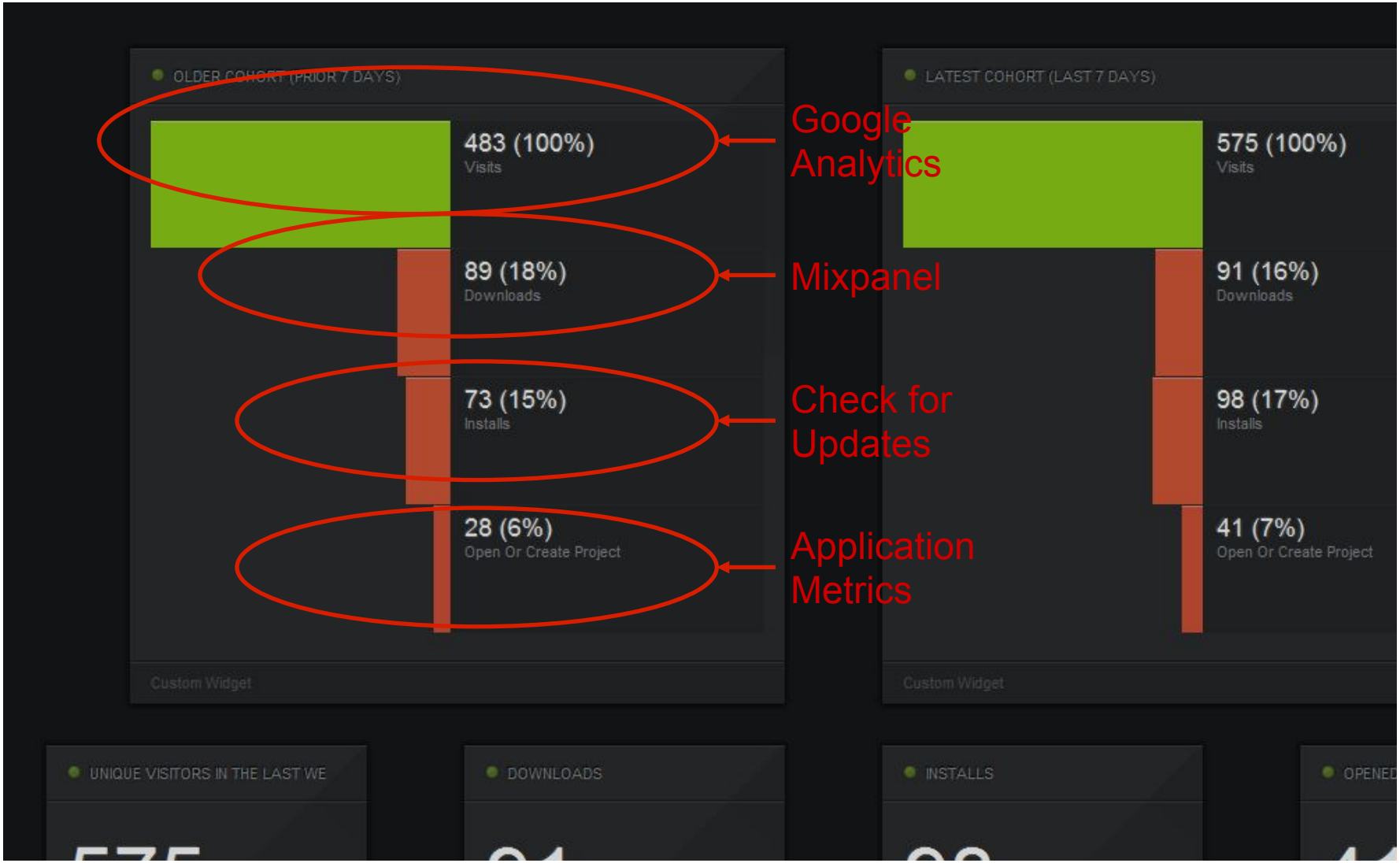
Measure your engine of growth

1. Paid: old customers cover the cost of acquiring new customers (eg dating business)
2. Viral: word of mouth (eg facebook)
3. Engagement or stickiness: (eg software leasing)

SQL Connect

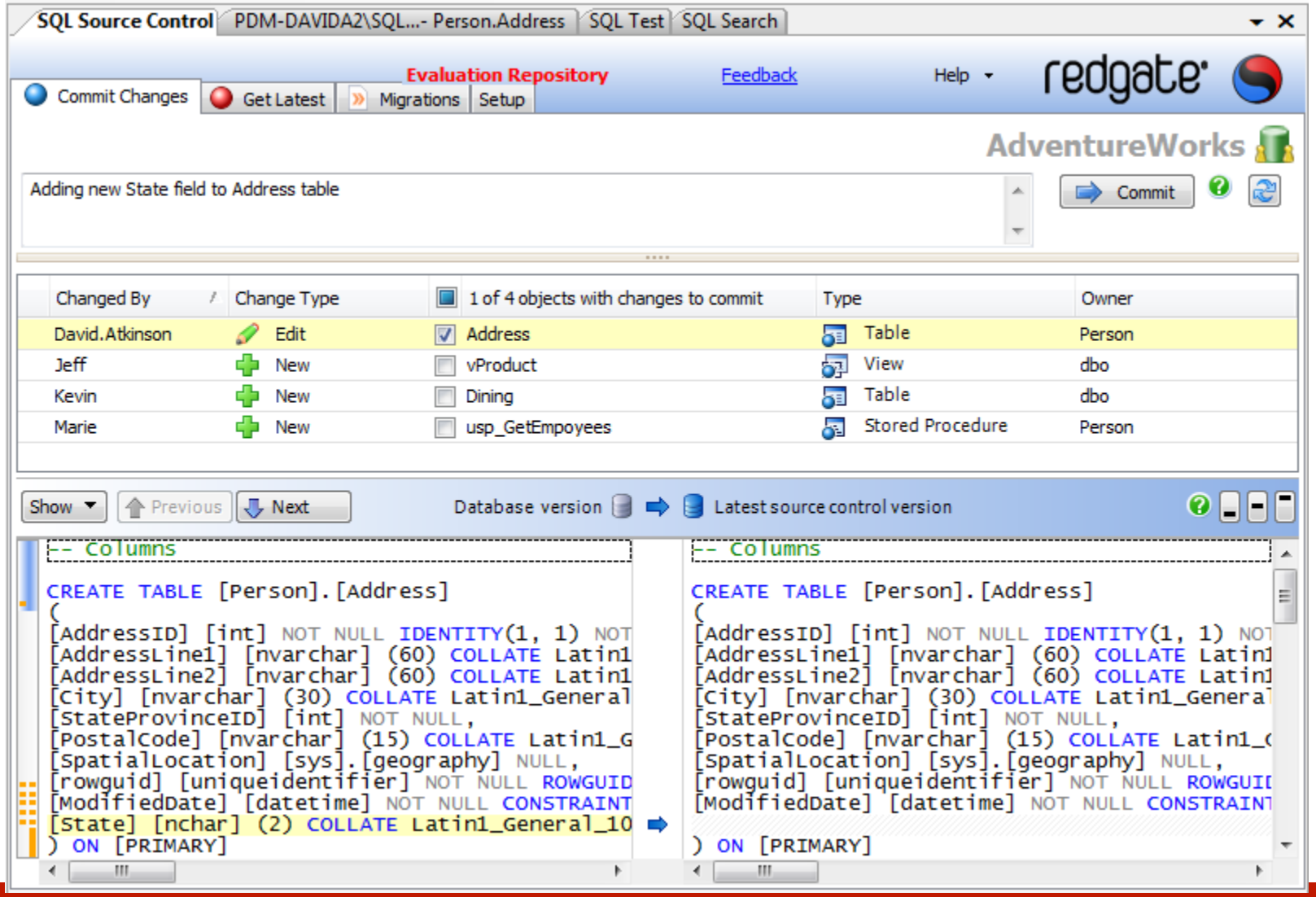

MyMVCWebApp - Microsoft Visual Studio
File Edit View Project Build Debug Team Data SQL Prompt 5 Tools Test Analyze Window Help
Debug
Any CPU
New Query - msts.te...l.Christofides (53))*
Connected to AdventureWorks on msts.testnet
Execute
Provide Feedback...
redgate
ALTER PROCEDURE [dbo].[uspGetContacts]
AS
SELECT Title ,
FirstName ,
LastName
FROM Person.Contact
1. Add a new project
The Red Gate Database Project lets you work connected to a database in real time, within Visual Studio.
2. Make a Change
Here, we've modified a stored procedure, and executed the SQL.
Visual Studio, with SQL Connect
100 %
Messages
Command(s) completed successfully.
4. Track changes
See changes in the Solution Explorer when you synchronize the project.
You can then commit and share changes with your existing source control system. Here we're using SVN and AnkhSVN.
Query executed successfully.
msts.testnet (10.50 RTM)
RED-GATE\Michael.Chris...
SQL Connect
Refresh
Synchronize
0 Conflicted Changes
0 Changes to the Connected Database
0 Changes to the Offline Creation Scripts
150 Identical Objects
3. Refresh and Synchronize
Review chages you've made and synchronize your database and project.
Provide Feedback...
redgate
SQL Connect
Pending Changes
Solution Explorer
Solution 'MyMVCWebApp' (2 projects)
AdventureWorks
Assemblies
Data
Database Triggers
Defaults
Extended Properties
Functions
Rules
Security
Service Broker
Storage
Stored Procedures
dbo.uspGetBillOfMaterials
dbo.uspGetContacts
dbo.uspGetEmployeeManagers
dbo.uspGetManagerEmployees
dbo.uspGetWhereUsedProductID
dbo.uspLogError
dbo.uspPrintError
HumanResources.uspUpdateEmployeeHireInfo
HumanResources.uspUpdateEmployeeLogin
HumanResources.uspUpdateEmployeePersona
Synonyms
Tables
Types
Views
RedGateDatabaseInfo
MyMVCWebApp
Properties
References
Account
App_Data
Scripts
Solution Explorer
Team Explorer
Ready
Ln 7
Col 5
Ch 5
INS

Monitoring the right metrics

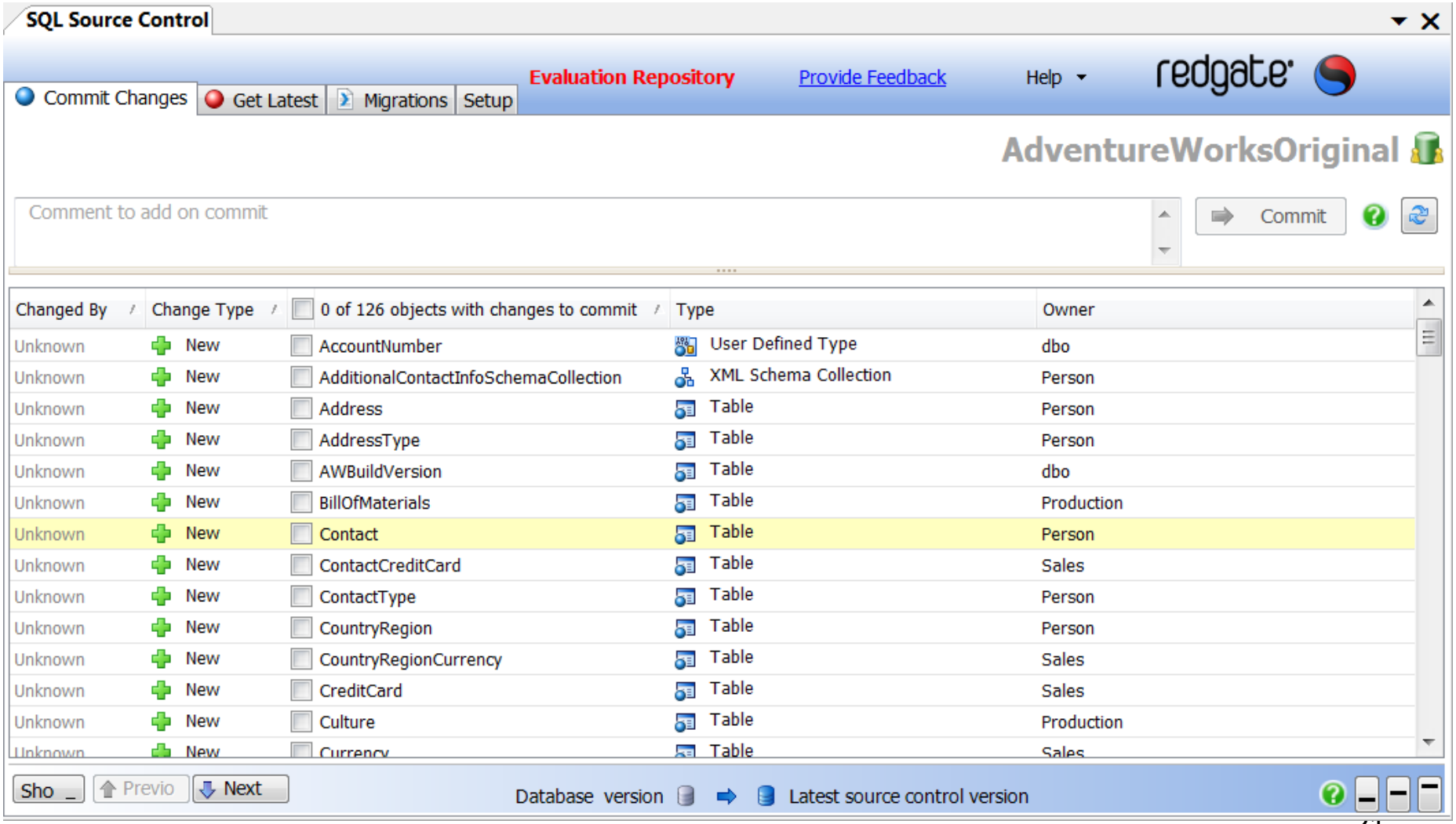


Measuring runtime behaviour

(and avoiding an expensive release)



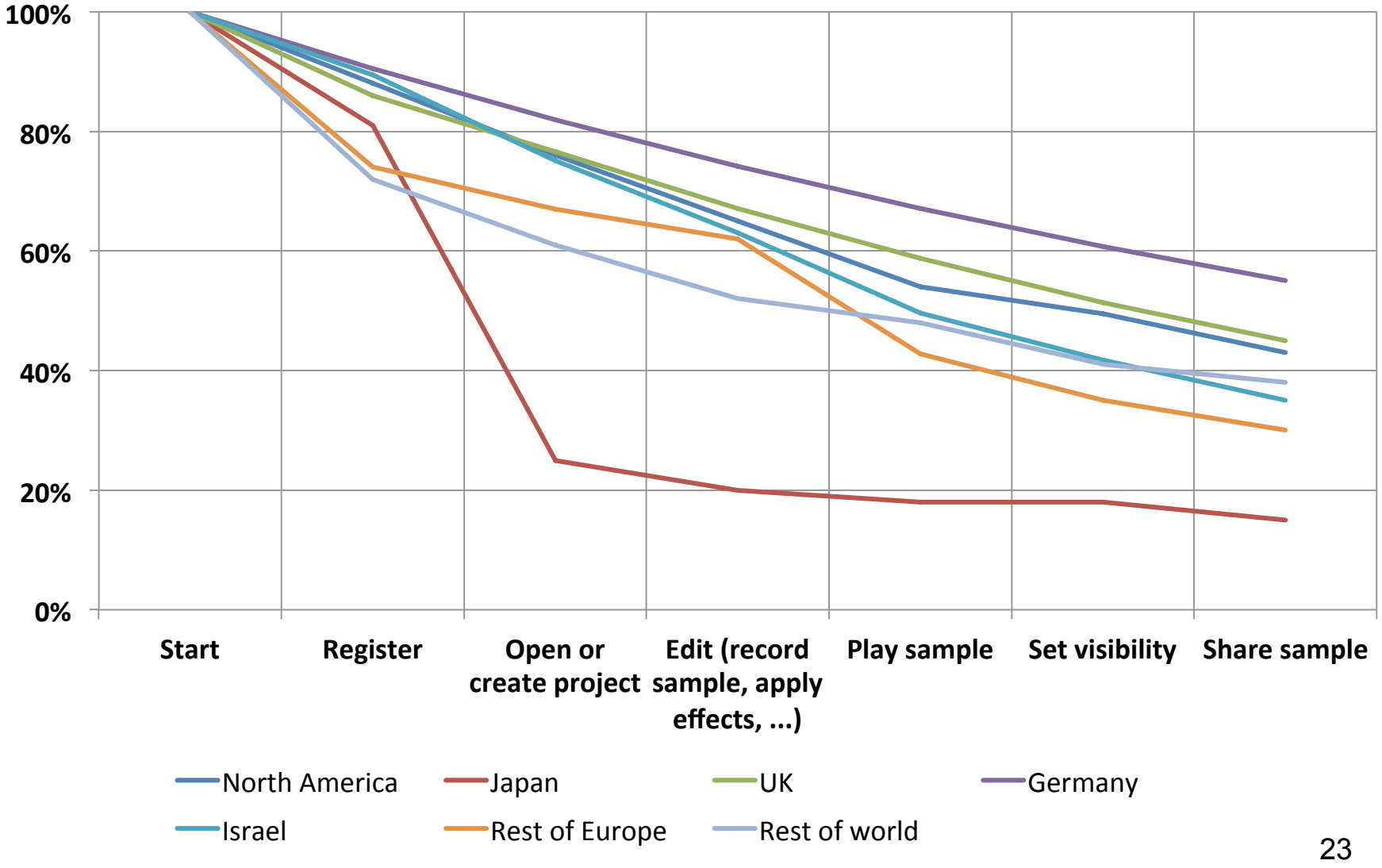
How quickly is the trace file overwritten?



Actions the team took

- Used ApplicationMetrics to measure the actual time before the trace file were overwritten
  - Were amazed users got less than 10 minutes
- Originally planned to use more trace files
  - Which clearly wouldn't work
- Instead the team chose to keep their own record of database changes
  - Which avoided an unnecessary release

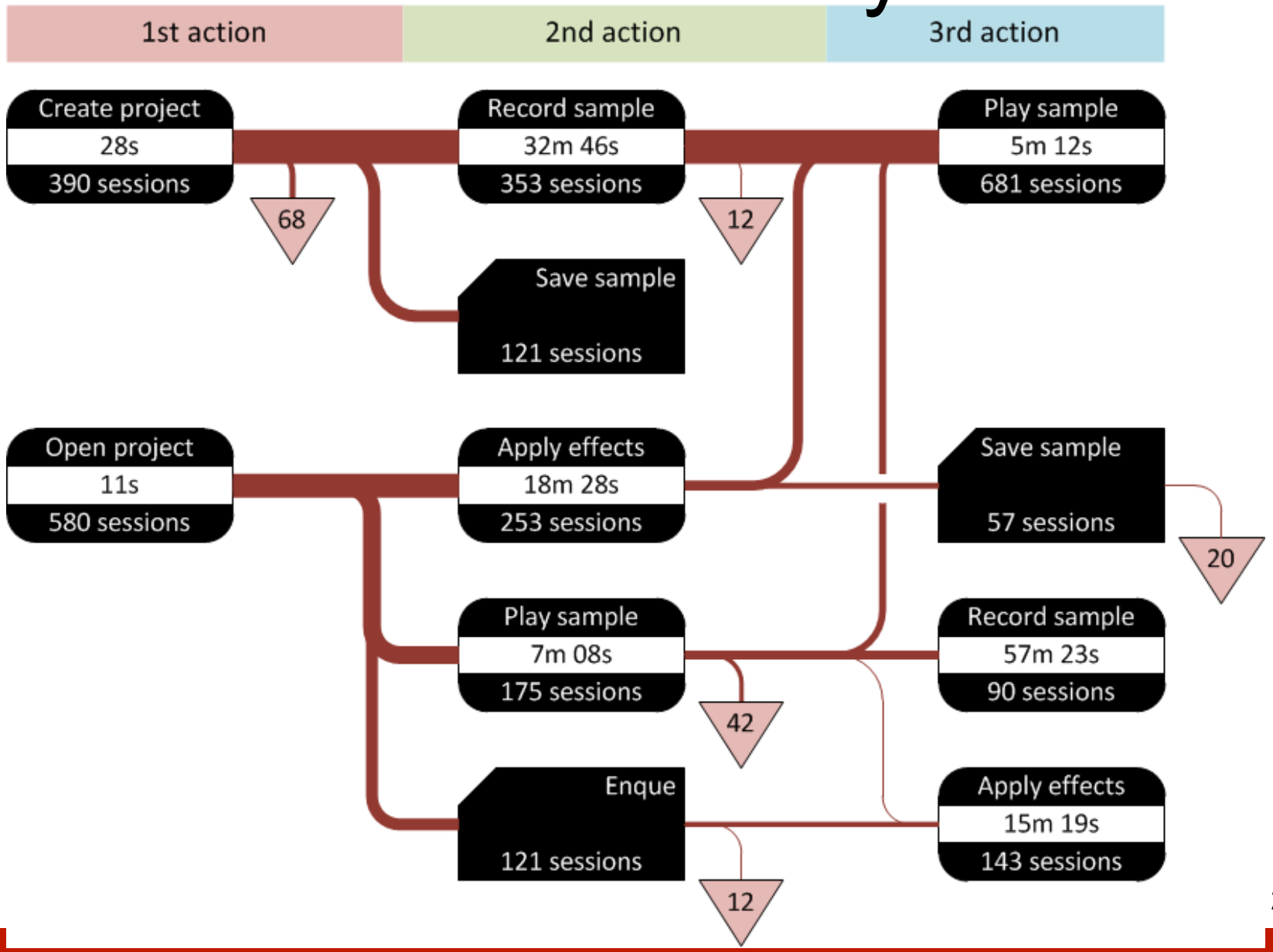
Monitoring usage in the field



First used in week	First time users	Percentage of users still using the product after:							
		1 week	2 weeks	3 weeks	4 weeks	5 weeks	6 weeks	7 weeks	8 weeks
47	5	60%	0%	20%	0%	0%	0%	0%	0%
48	4	50%	75%	0%	0%	0%	0%	0%	0%
49	8	75%	50%	25%	25%	13%	25%	0%	0%
50	12	42%	25%	8%	17%	8%	8%	0%	0%
51	15	53%	40%	20%	0%	0%	0%	0%	0%
52	48	63%	25%	21%	6%	2%	4%	2%	0%
53	42	83%	69%	62%	55%	29%	19%	0%	0%
01	45	80%	60%	44%	22%	18%	11%	9%	
02	62	50%	37%	19%	8%	3%	0%		
03	76	84%	75%	59%	47%	41%			
04	69	75%	70%	57%	49%				
05	149	60%	36%	20%					
06	184	72%	47%						
07	167	50%							
08	156								

Retention analysis for the Demon Test Coverage tool

Task flow analysis

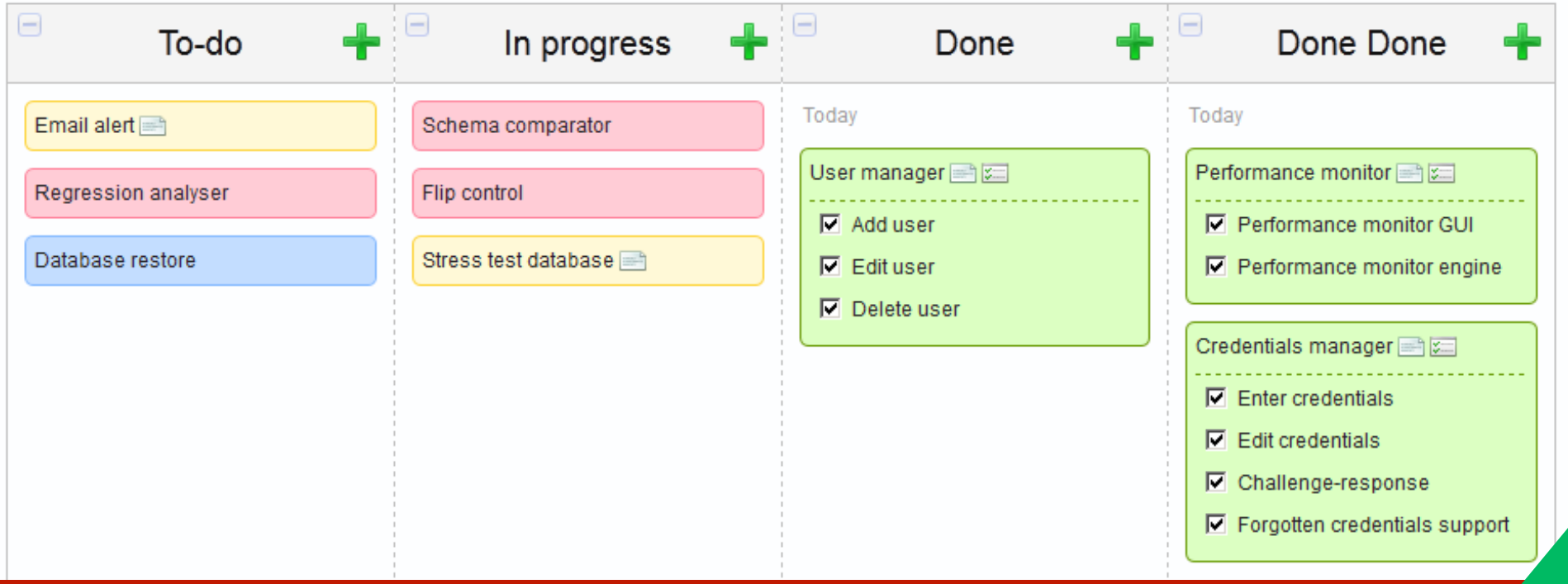


"Done Done"

ie the goals of the change have been achieved

- ALM integration
- Linking to a Kanban

- Choose an appropriate definition, eg
  - Feature is used at least twice by 5 users



Tools for runtime analytics

ApplicationMetics

- Our Runtime Analytics whitepaper
- If you want to follow our progress or start using ApplicationMetrics (for free)
  - www.applicationmetrics.com