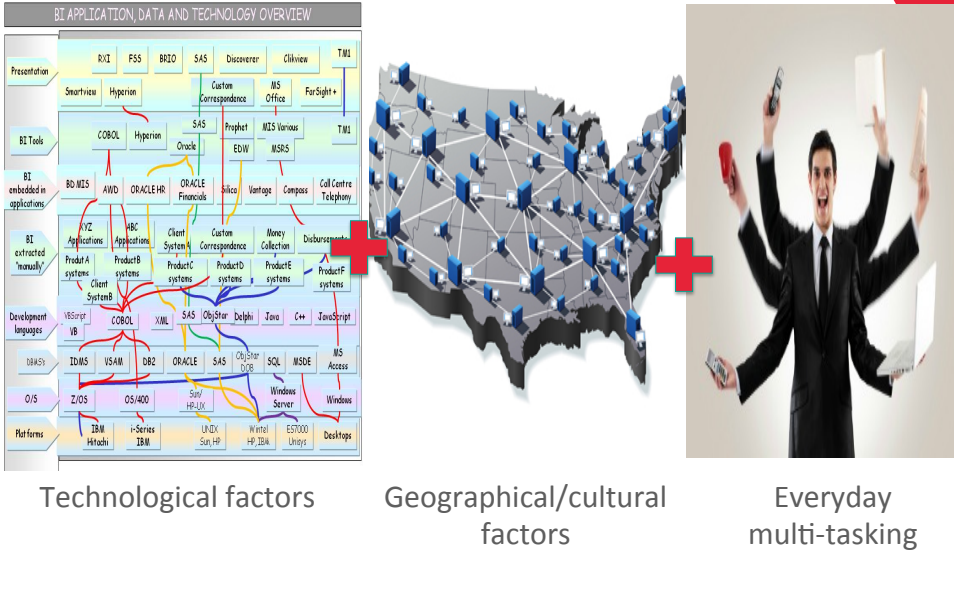Redmind

Avoiding Vendor-Driven Delivery
Julie Gardiner

Email: julie@qstc.co.uk
Twitter ID - @cheekytester

How well do we communicate?

Complexity factors

Technological factors

Geographical/cultural factors

Everyday multi-tasking

The Manager's view

- Create IT solutions = add business value, effective, easy to maintain and lean.
- Cost of purchasing software < the additional workload in-house – buy!
- What about future costs?
- Where are we in making this decision?
  - Where's the quality view?

Developers time

What is vendor-driven delivery?
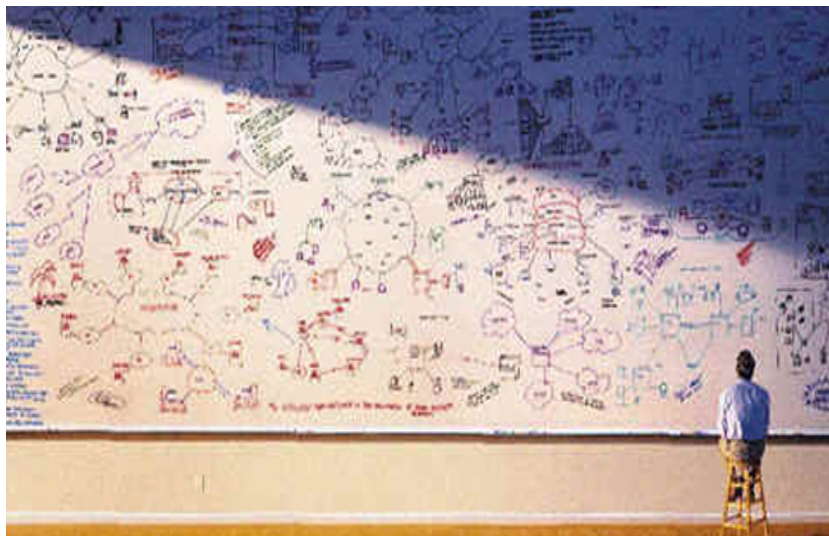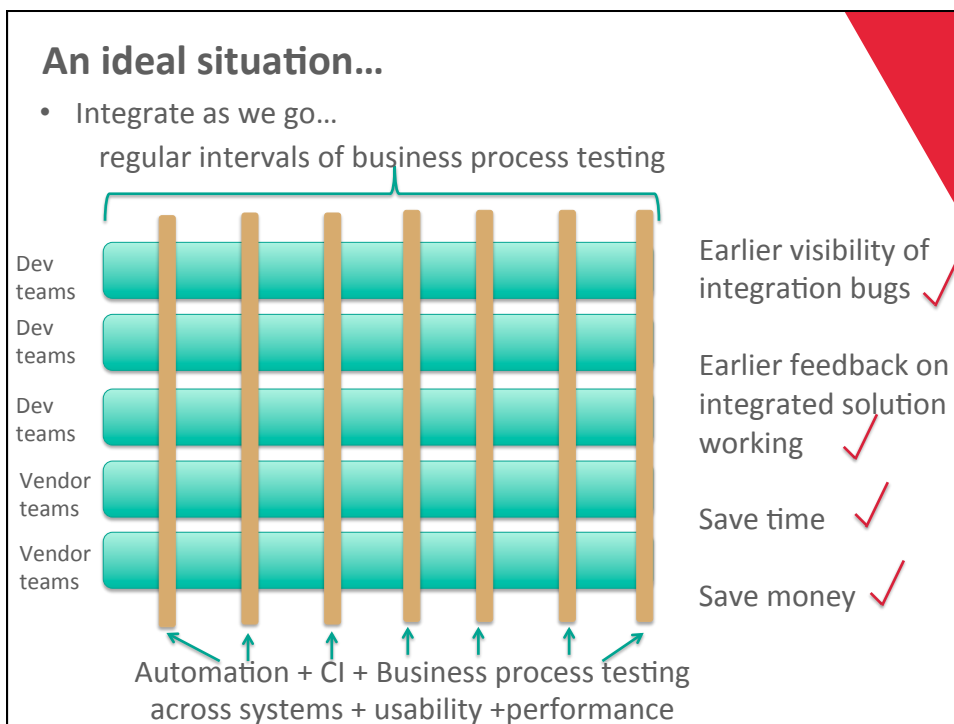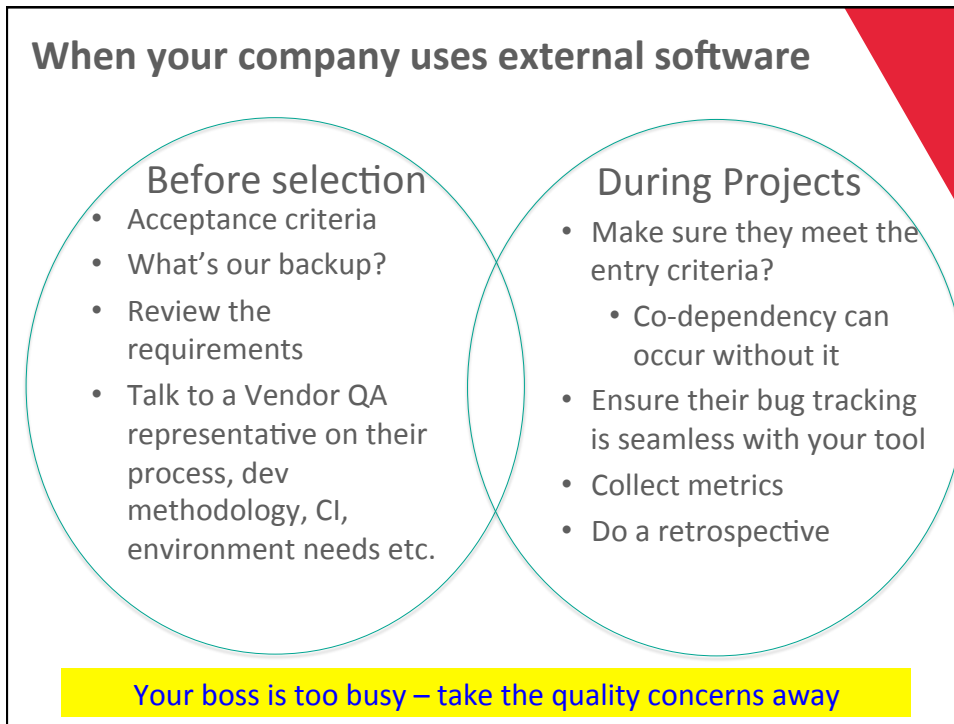

IDENTIFY NEED   BUILD NEED   DELIVER VALUE

WHO MAKES THIS DECISION?

Business & technical communication is needed

Mapping Business processes to technical structure is needed
Impact analysis & priority to the business

When your company uses external software

Before selection
- Acceptance criteria
- What's our backup?
- Review the requirements
- Talk to a Vendor QA representative on their process, dev methodology, CI, environment needs etc.

During Projects
- Make sure they meet the entry criteria?
  - Co-dependency can occur without it
- Ensure their bug tracking is seamless with your tool
- Collect metrics
- Do a retrospective

Your boss is too busy – take the quality concerns away

An ideal situation…

- Integrate as we go…

regular intervals of business process testing

Dev teams

Dev teams

Dev teams

Vendor teams

Vendor teams

Earlier visibility of integration bugs ✓

Earlier feedback on integrated solution working ✓

Save time ✓

Save money ✓

Automation + CI + Business process testing across systems + usability +performance
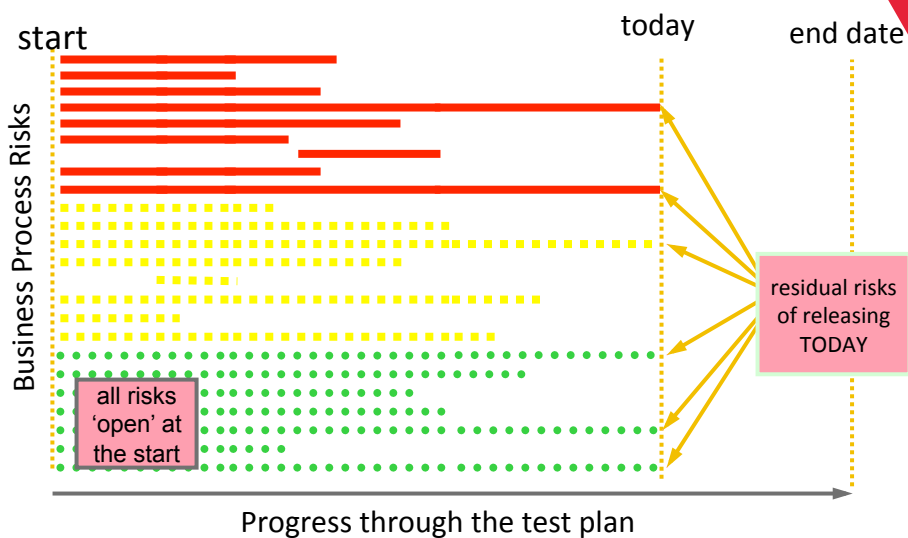
But what if we're too far away from this

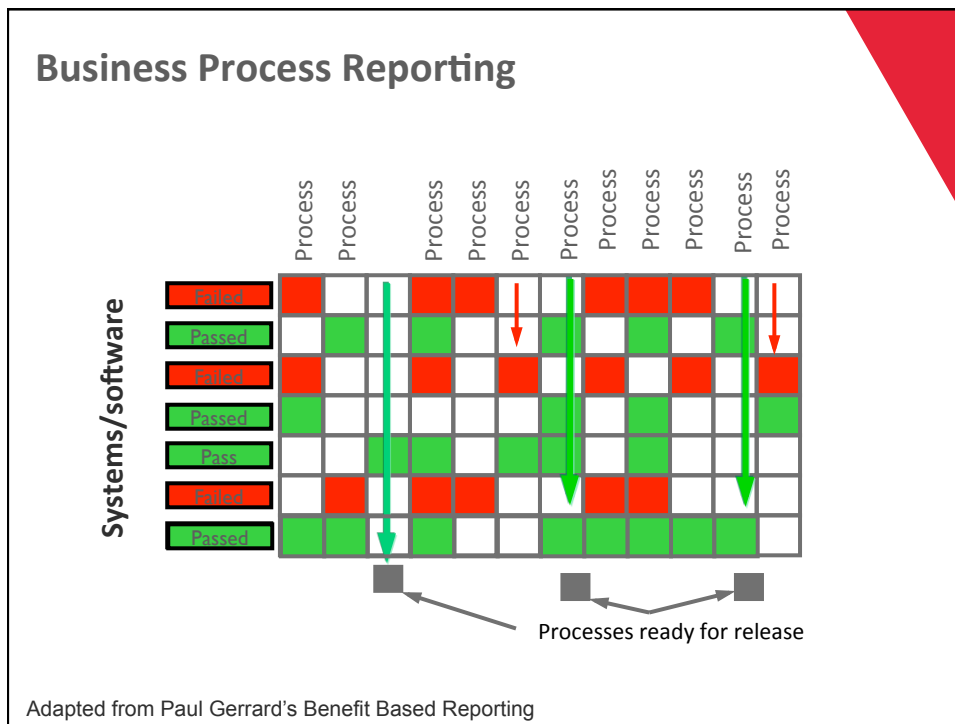Show what we can do for our company

- Show the value by:
  - reporting how much we could save the company
    - How many bugs are you finding that are not related to UAT specific testing or System Integration Test?
    - use the money to improve environments, automation etc.
  - demonstrating test effectiveness
  - use language business & managers understand
    - risk rules – test cases don't!

cost saving
per bug

Defect Detection
Percentage   (DDP)

risk

Risk-based reporting

start

today

end date

Business Process Risks

residual risks
of releasing
TODAY

all risks
'open' at
the start

Progress through the test plan
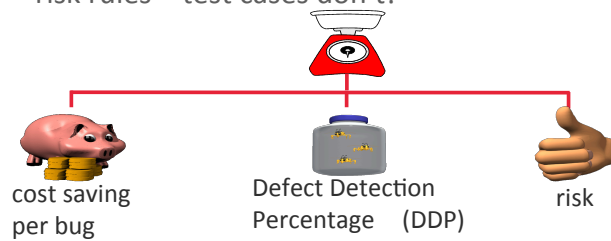
Source: Paul Gerrard & Neil Thompson

Business Process Reporting

Processes ready for release

Adapted from Paul Gerrard's Benefit Based Reporting

Project 1

Background

- Financial company
  - New product to complement existing solution
- Timing of my contribution

Lessons learned

- Friends don't always make the best partners in business
- Different viewpoint and environment
- Trust your instincts
- The right decision can be made if you give the right information- risk ruled!
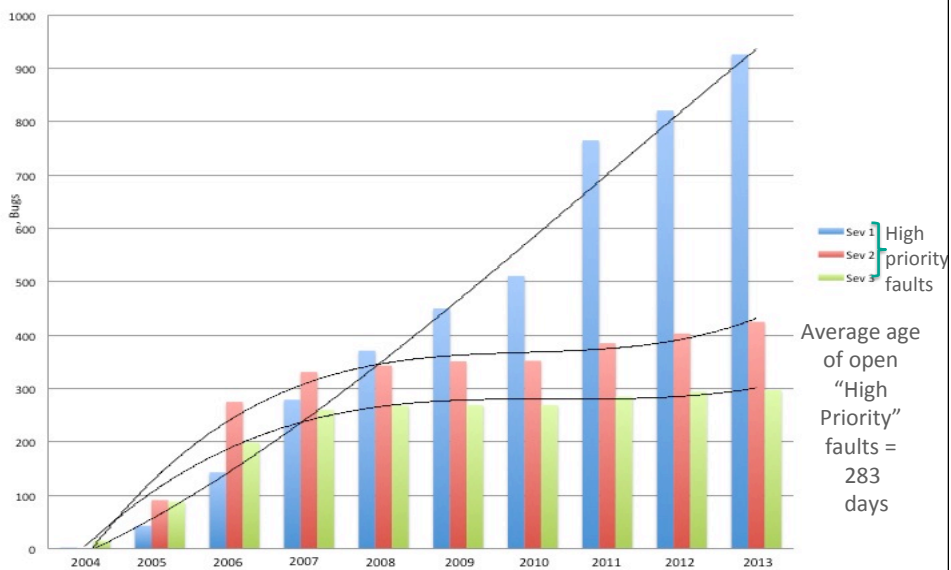- Review your testing regularly

End result? Not released.
Asked to provide acceptance criteria into contracts going forward
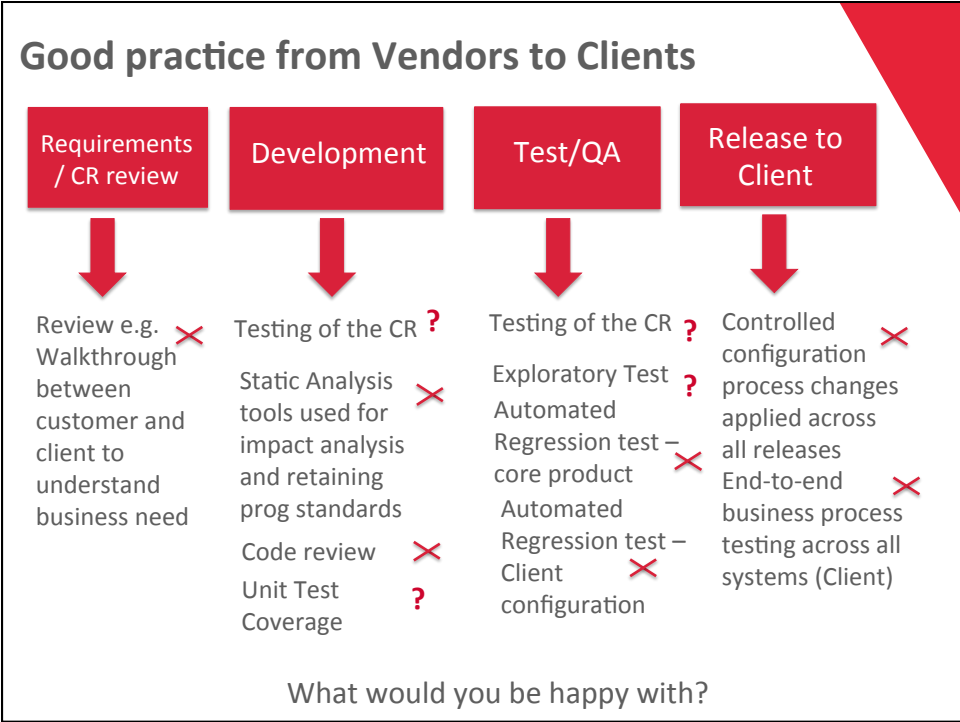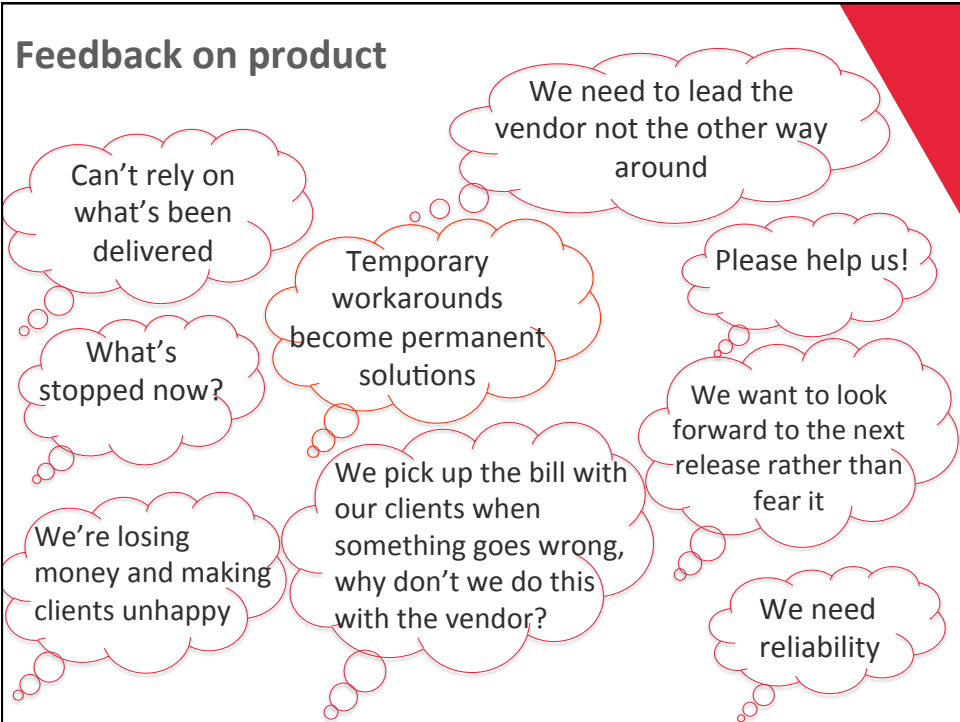
Project 2

Background

- Media company
- Contract acceptance criteria in place including penalty payments
- All software was created externally
- Desire to improve after terrible release
- Requirements produced but were never right

Bug Trend in the product

Sev 1 } High
Sev 2 } priority
Sev 3 } faults

Average age of open "High Priority" faults = 283 days

Feedback on product

We need to lead the vendor not the other way around

Can't rely on what's been delivered

Temporary workarounds become permanent solutions

Please help us!

What's stopped now?

We want to look forward to the next release rather than fear it

We pick up the bill with our clients when something goes wrong, why don't we do this with the vendor?

We're losing money and making clients unhappy

We need reliability

Good practice from Vendors to Clients

Requirements / CR review	Development	Test/QA	Release to Client
Review e.g. Walkthrough between customer and client to understand business need ✕	Testing of the CR ? Static Analysis tools used for impact analysis and retaining prog standards ✕ Code review ✕ Unit Test Coverage ?	Testing of the CR ? Exploratory Test ? Automated Regression test – core product ✕ Automated Regression test – Client configuration ✕	Controlled configuration process changes applied across all releases ✕ End-to-end business process testing across all systems (Client) ✕

What would you be happy with?

Project 2 - What did we do?

Background
- Media company
- Contract acceptance criteria in place including penalty payments
- All software was created externally
- Desire to improve after terrible release
- Requirements produced but were never right

What did we do?
- Mapped Business processes to the systems
- Sorted our testing first
- Tried to influence vendor testing
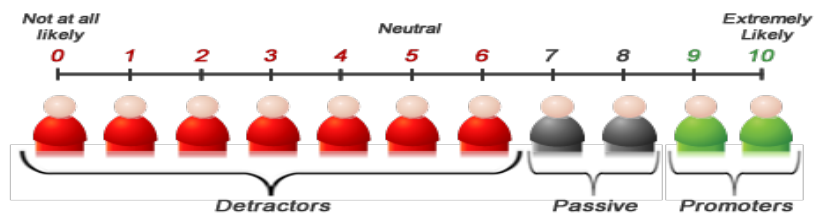- Enforce entry criteria
- Business reporting

How many high priority bugs found in production after 4 weeks?

Guidelines for you…

Test doc. available
Exploratory testing
Strategy, schedule
Test tools, approach
Evidence of testing
Reporting rules
Show the savings

Monetary consequences
Automated tests- core and customised
Gather info on company
Integrated Business Tests
Contract Acceptance criteria

Requirements understood & agreed
Impact analysis established
Non-functional testing considered
**G**reat Test environments
Static analysis performed

Other ideas

- Let's have our own tripadvisor for vendors and opensource products
  - comparethevendors.com?
- Vendor NPS

Not at all likely
Neutral
Extremely Likely

0 1 2 3 4 5 6 7 8 9 10

Detractors   Passive   Promoters

Net Promoter Score = % promoters - % Detractors

- Lets help build a community for testers using external software

---

Final thoughts – Time for Action

Now

You can change

Others can help you with
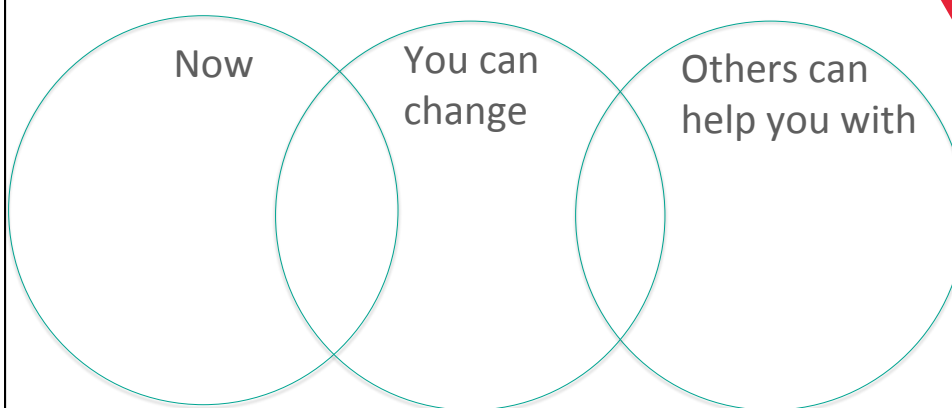
Email: julie@redmind.se
Twitter ID: @cheekytester