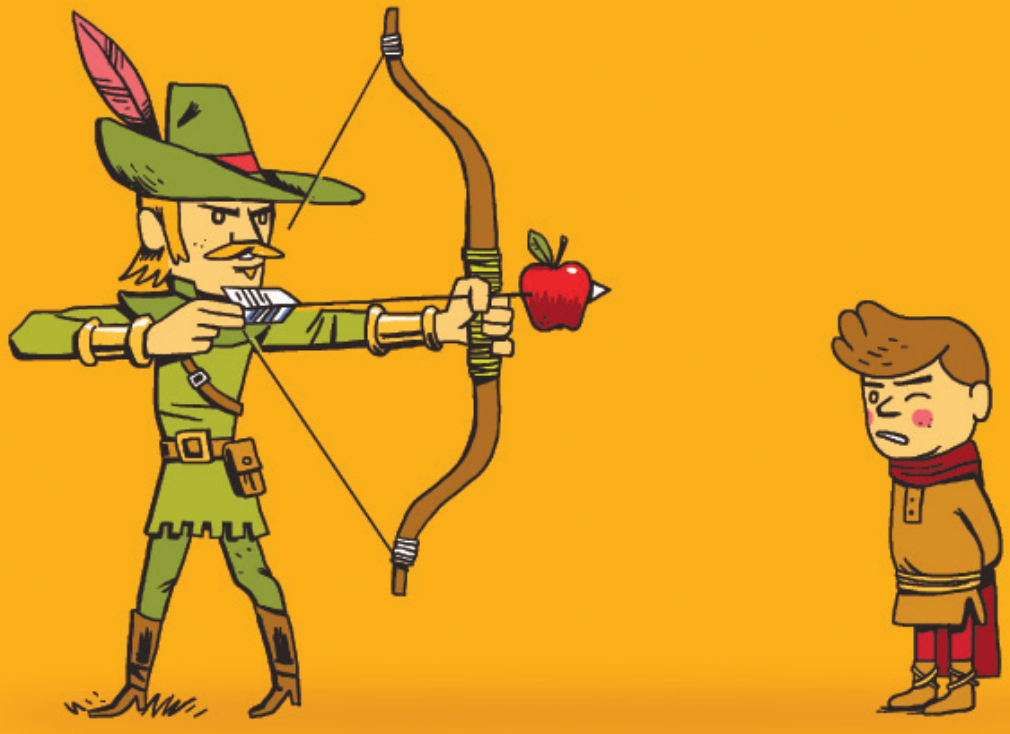

FIFTY
Quick
IDEAS

FIFTY QUICK IDEAS TO IMPROVE YOUR USER STORIES

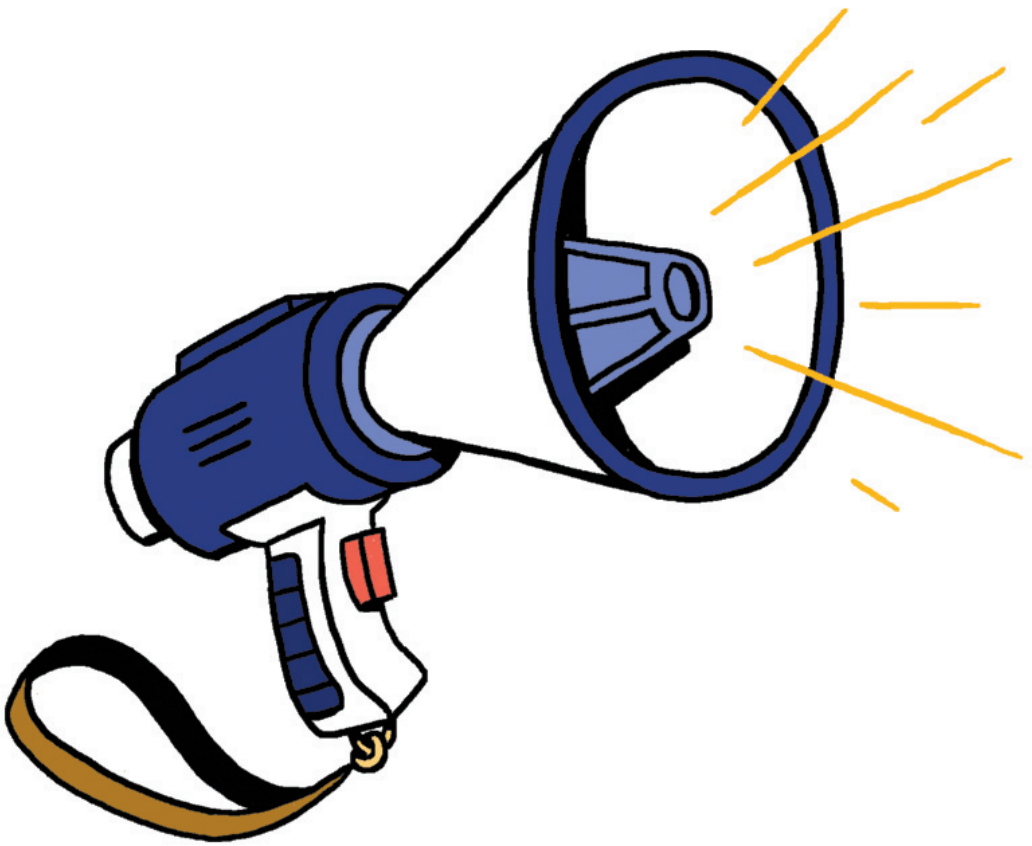


by Gojko Adzic and David Evans


FIFTY
Quick
IDEAS

TELL STORIES, DON'T WRITE THEM

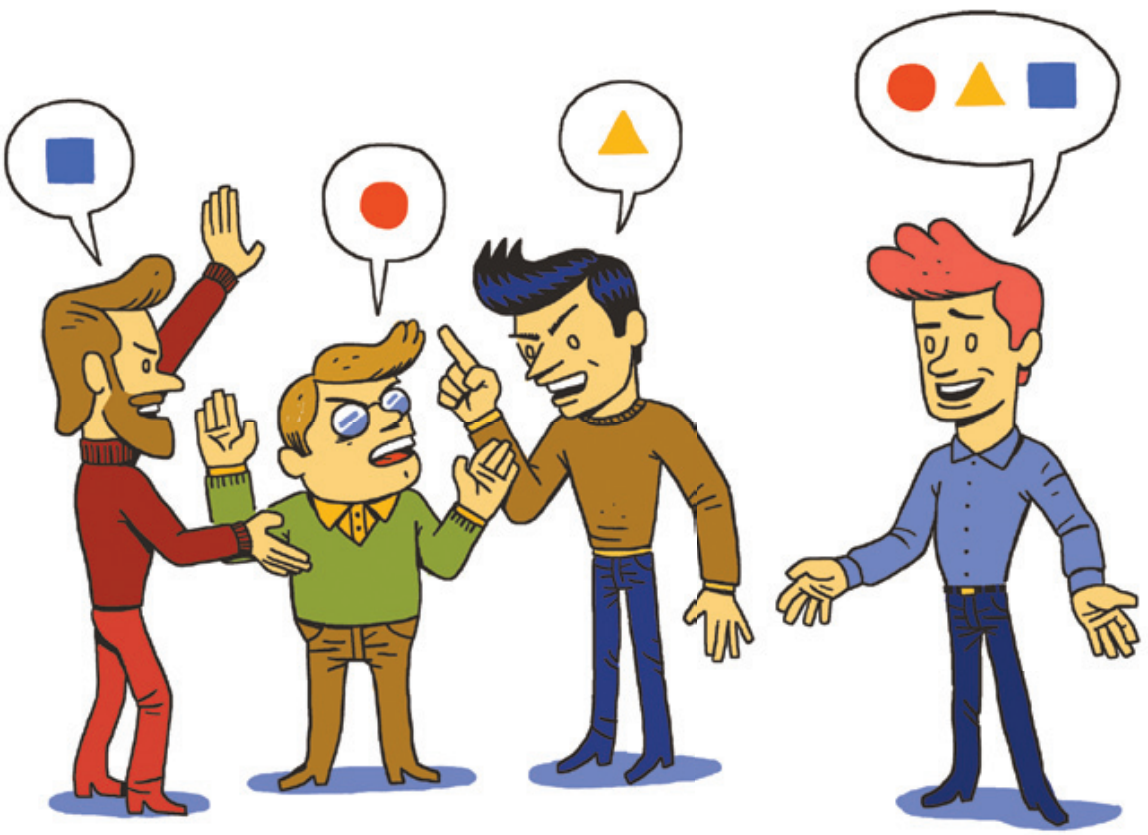
Collaborate to identify requirements instead of handing over documentation




FIFTY
Quick
IDEAS

DON'T WORRY TOO MUCH ABOUT STORY FORMAT

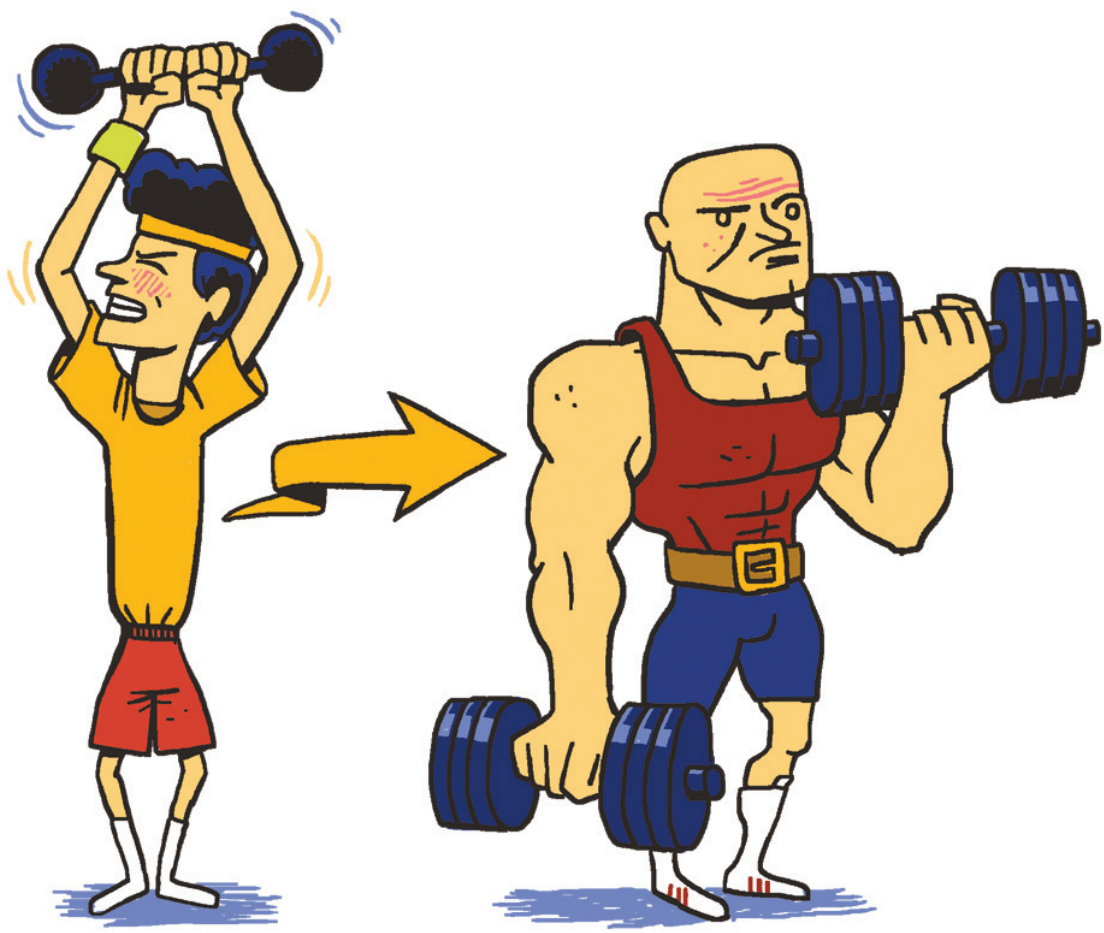
Collaborate to identify requirements instead of handing over documentation




FIFTY
Quick
IDEAS

DESCRIBE A BEHAVIOUR CHANGE

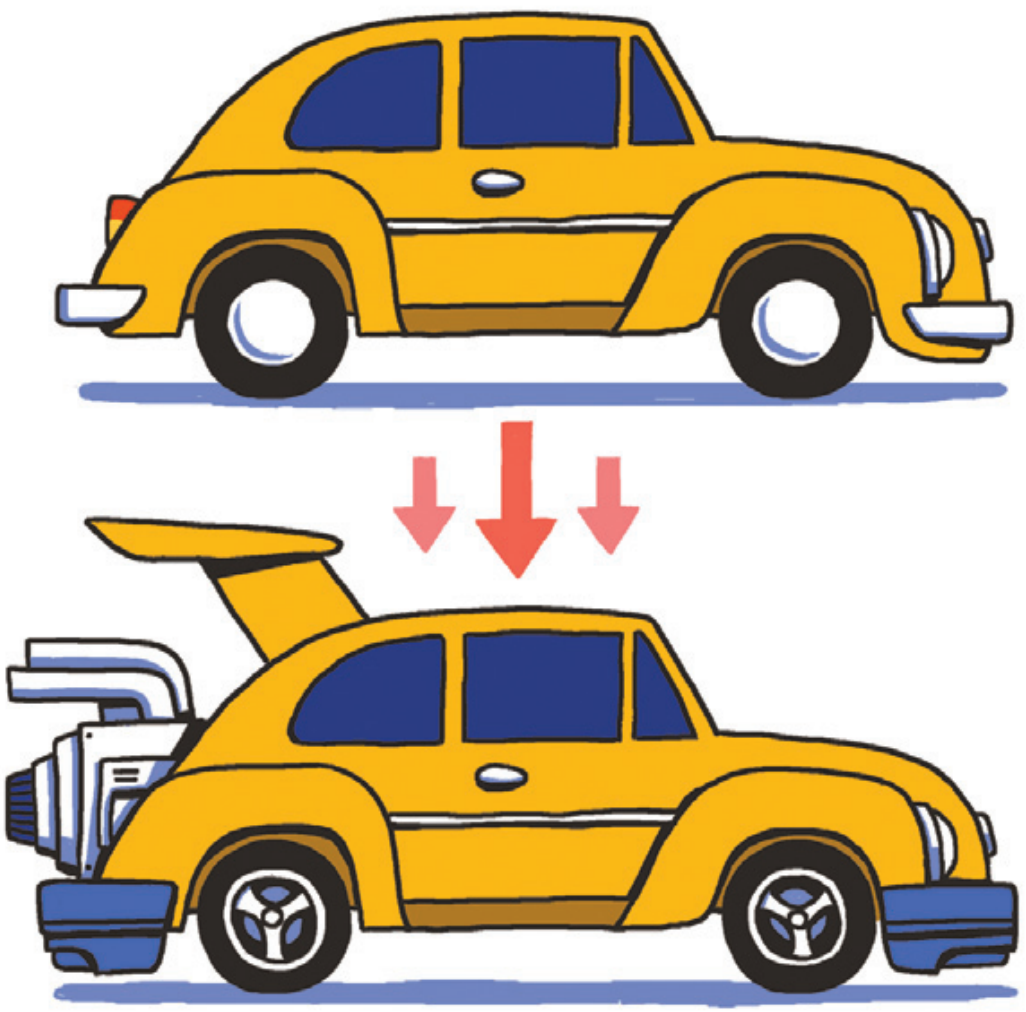
Capture a behaviour change to make a story measurable from a business perspective




FIFTY
Quick
IDEAS

DESCRIBE THE SYSTEM CHANGE

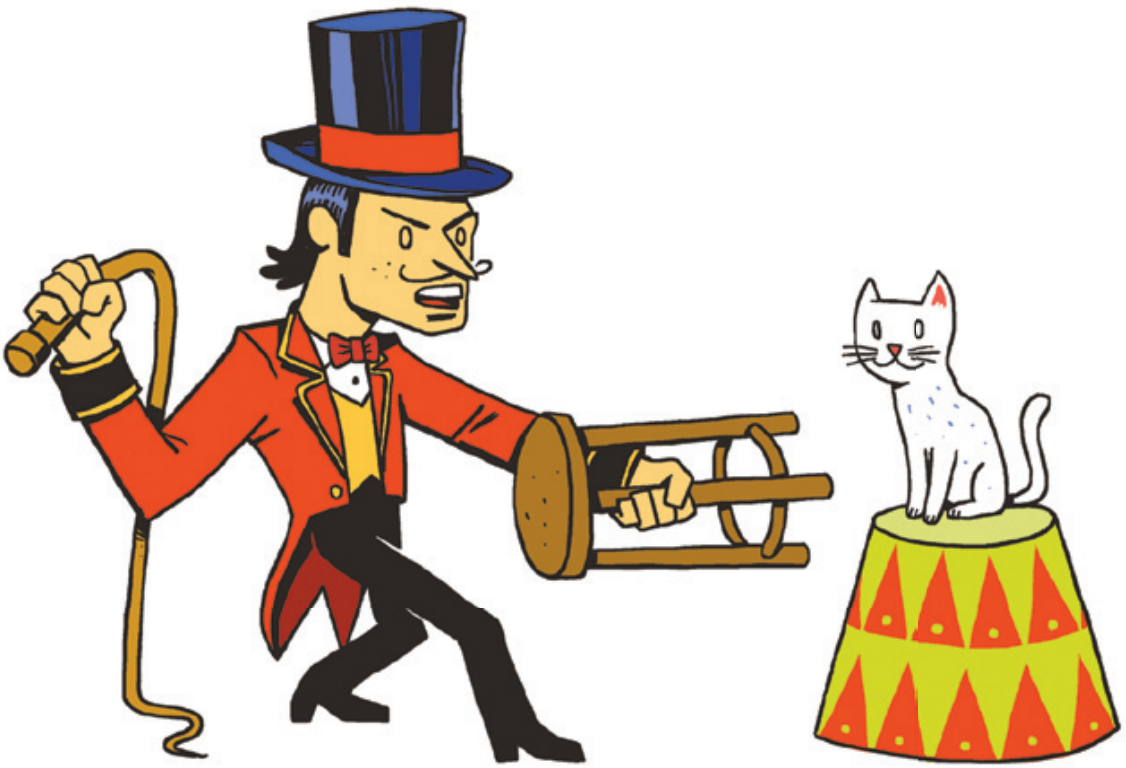
Clearly delineate the scope of the change to help create shared understanding




FIFTY
Quick
IDEAS

APPROACH STORIES AS SURVIVABLE EXPERIMENTS

Stories are based on assumptions that might turn out to be right or wrong




FIFTY
Quick
IDEAS

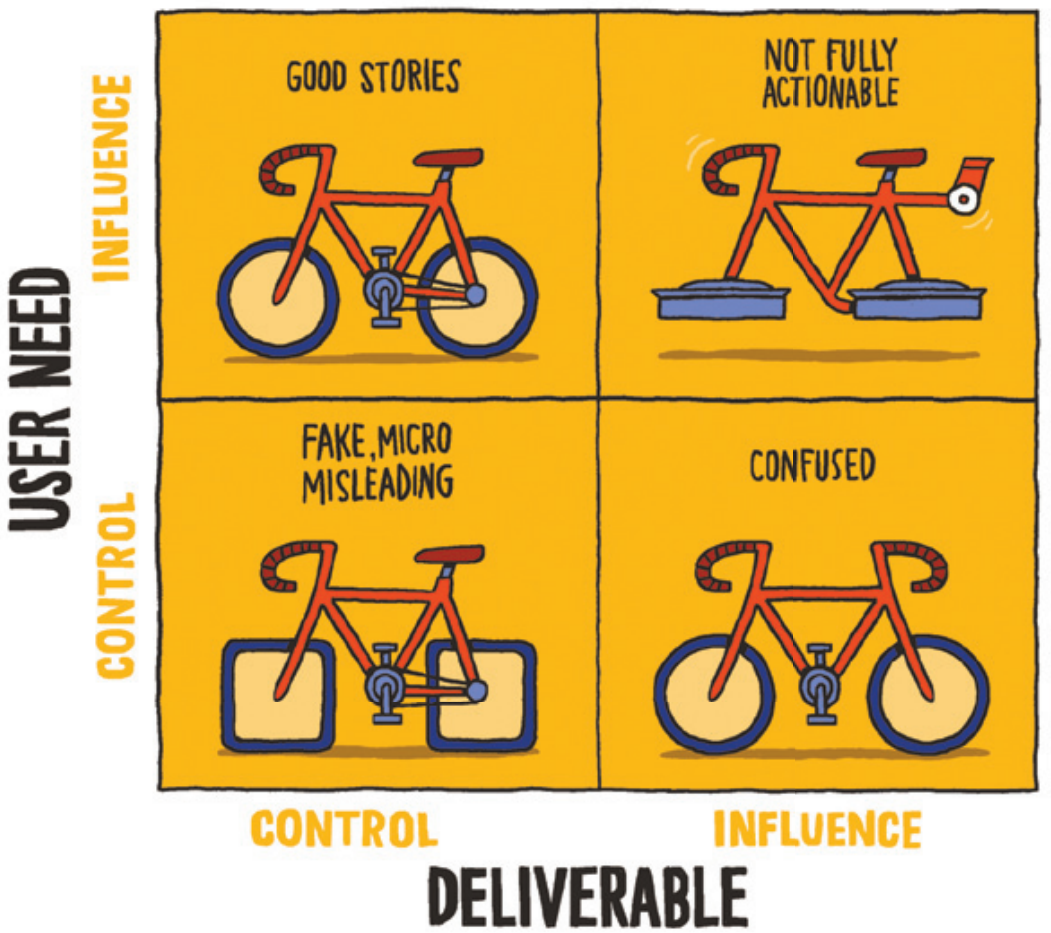
WATCH OUT FOR GENERIC ROLES

Focus on a specific customer role to help provide a useful context for discussion


FIFTY
Quick
IDEAS

EVALUATE ZONE OF CONTROL AND SPHERE OF INFLUENCE

Deliverables should be in your zone of control, and user needs in your sphere of influence




FIFTY
Quick
IDEAS

PUT A ‘BEST BEFORE’ DATE ON STORIES

Manage time-constrained stories separately, so they don’t turn into emergencies


FIFTY
Quick
IDEAS

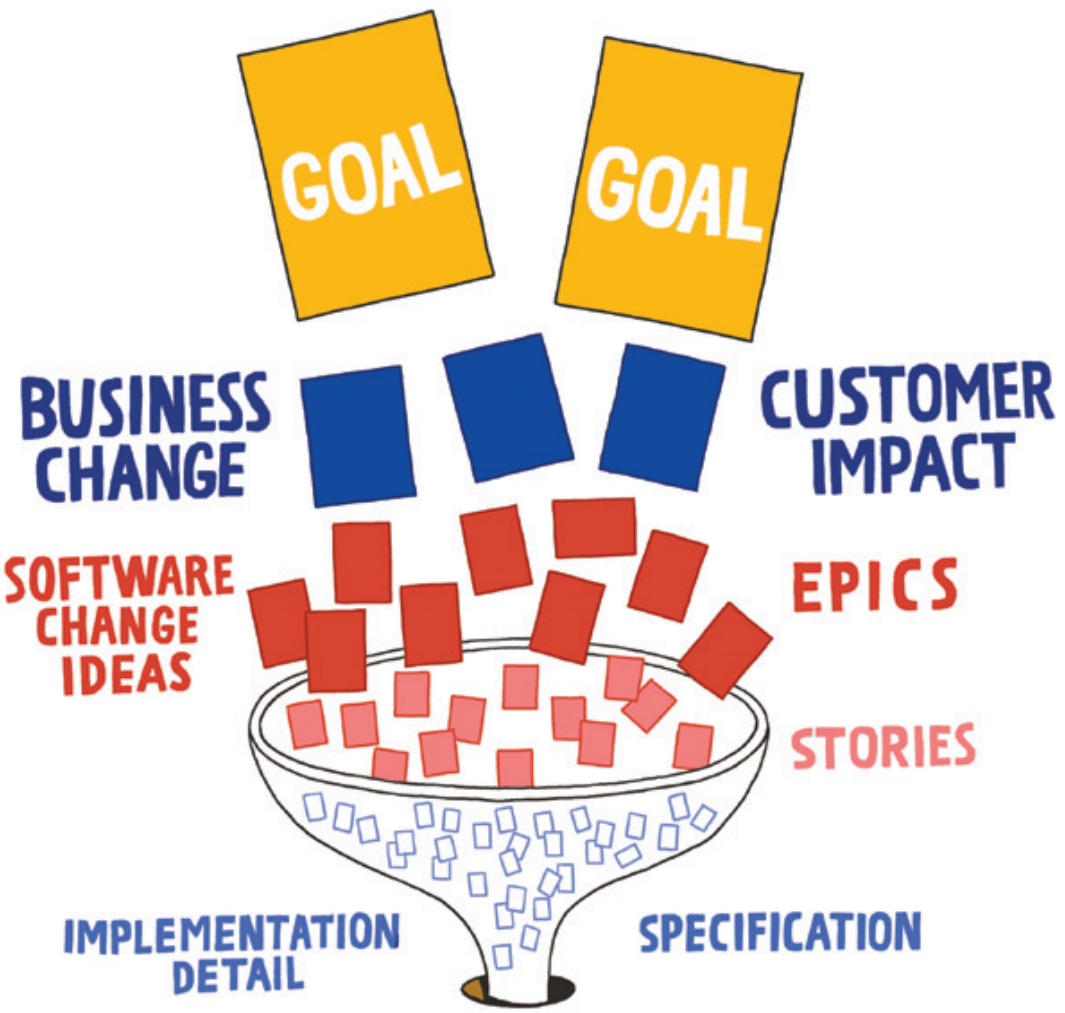
SET DEADLINES FOR ADDRESSING MAJOR RISKS

Strike the right balance between short-term business wins and long-term sustainability


FIFTY
Quick
IDEAS

USE HIERARCHICAL BACKLOGS

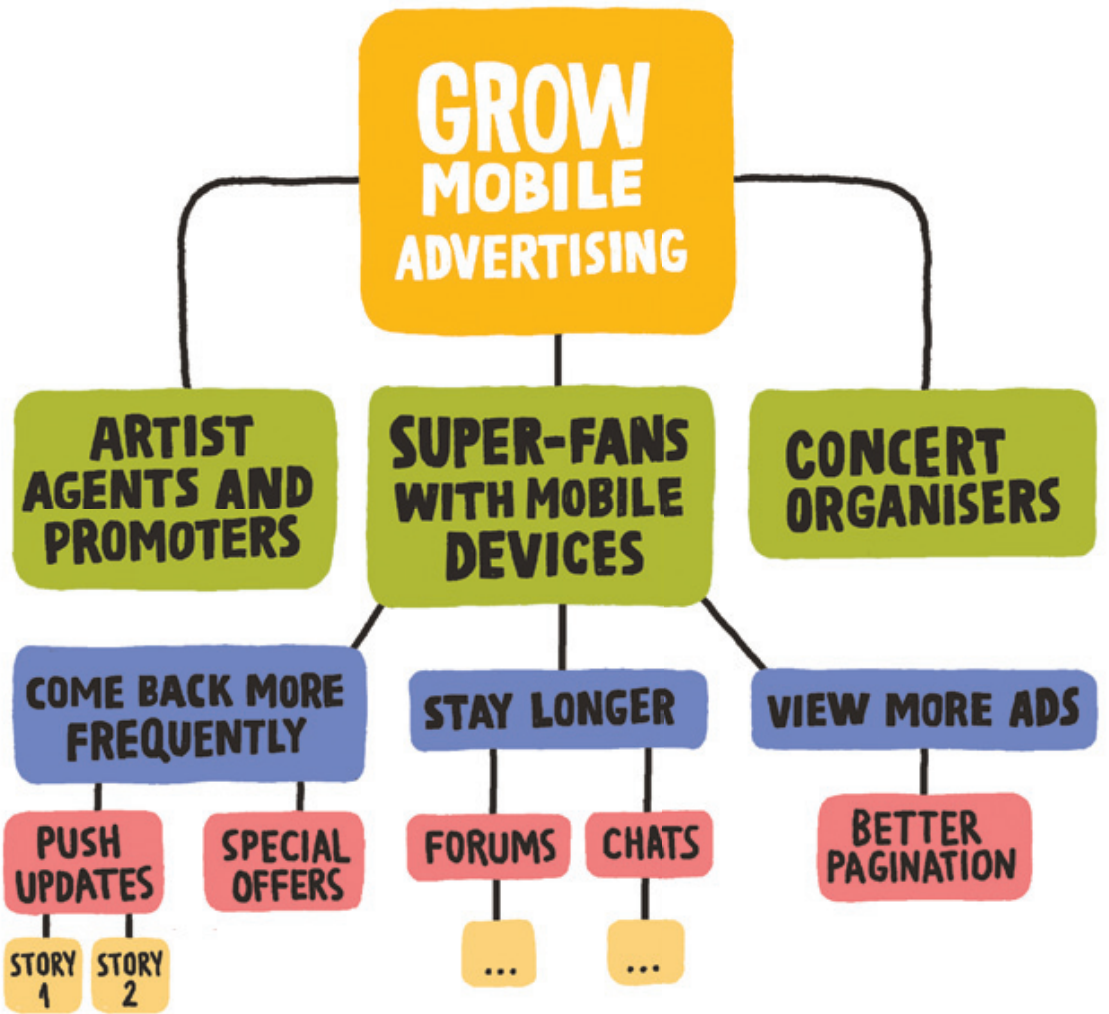
Hierarchical plans allow organisations to react effectively to changing market opportunities




FIFTY
Quick
IDEAS

GROUP STORIES BY IMPACT

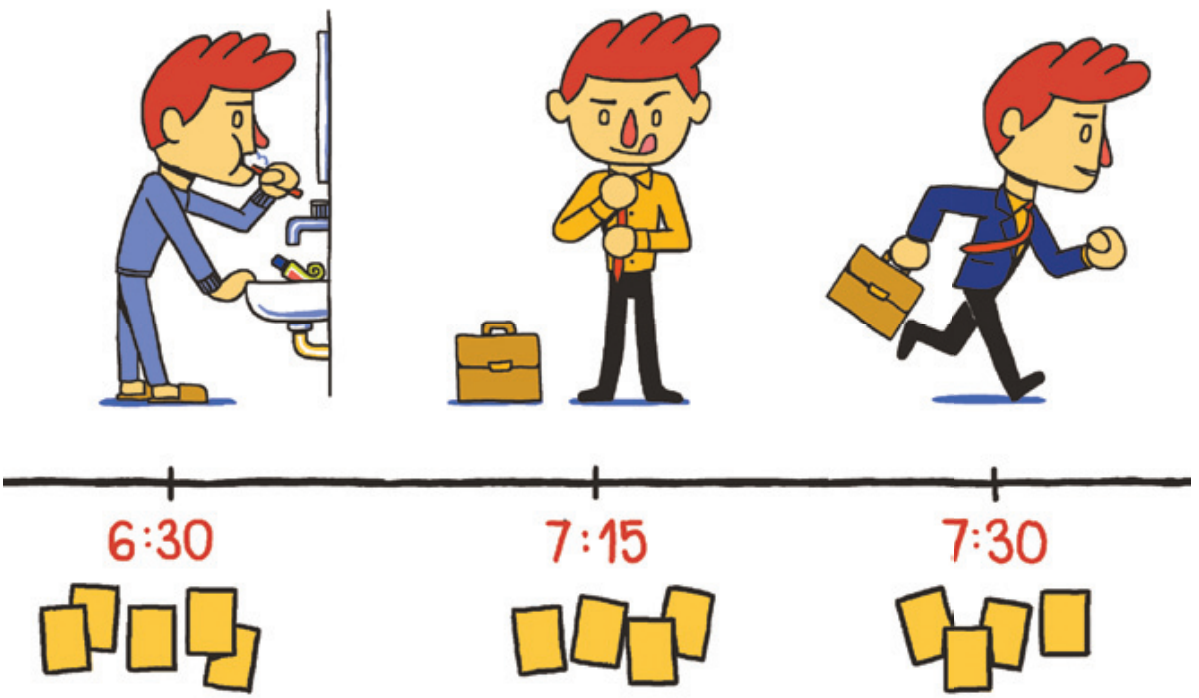
Visualise the connection between goals and deliverables to help stakeholders align on priorities




FIFTY
Quick
IDEAS

CREATE A USER STORY MAP

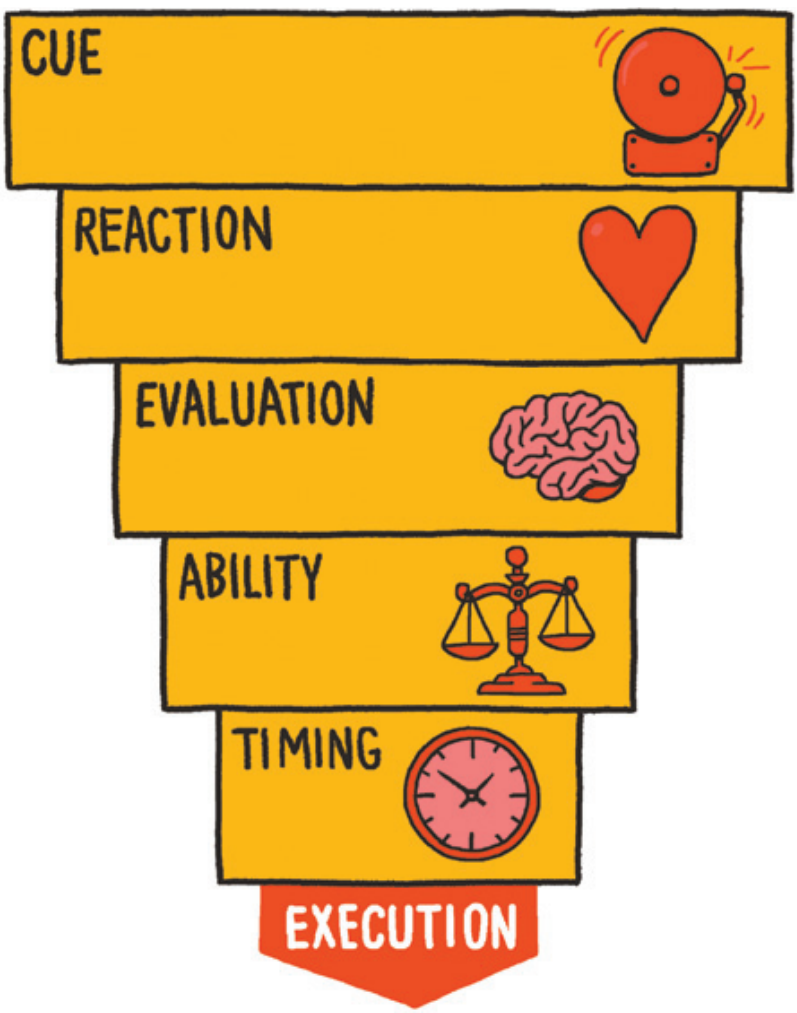
Remember that the customer experience often starts and ends outside interaction with software




FIFTY
Quick
IDEAS

CHANGE BEHAVIOURS USING THE CREATE FUNNEL

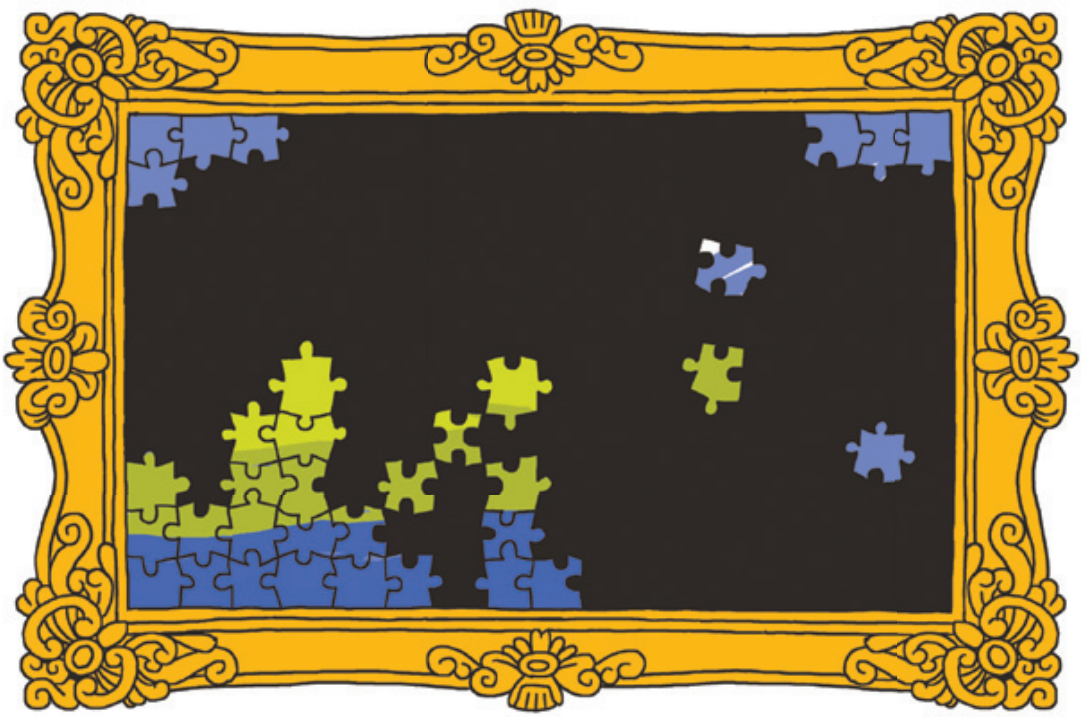
Break activities into CREATE stages to come up with great product ideas




FIFTY
Quick
IDEAS

SET OUT GLOBAL CONCERNS AT THE START OF A MILESTONE

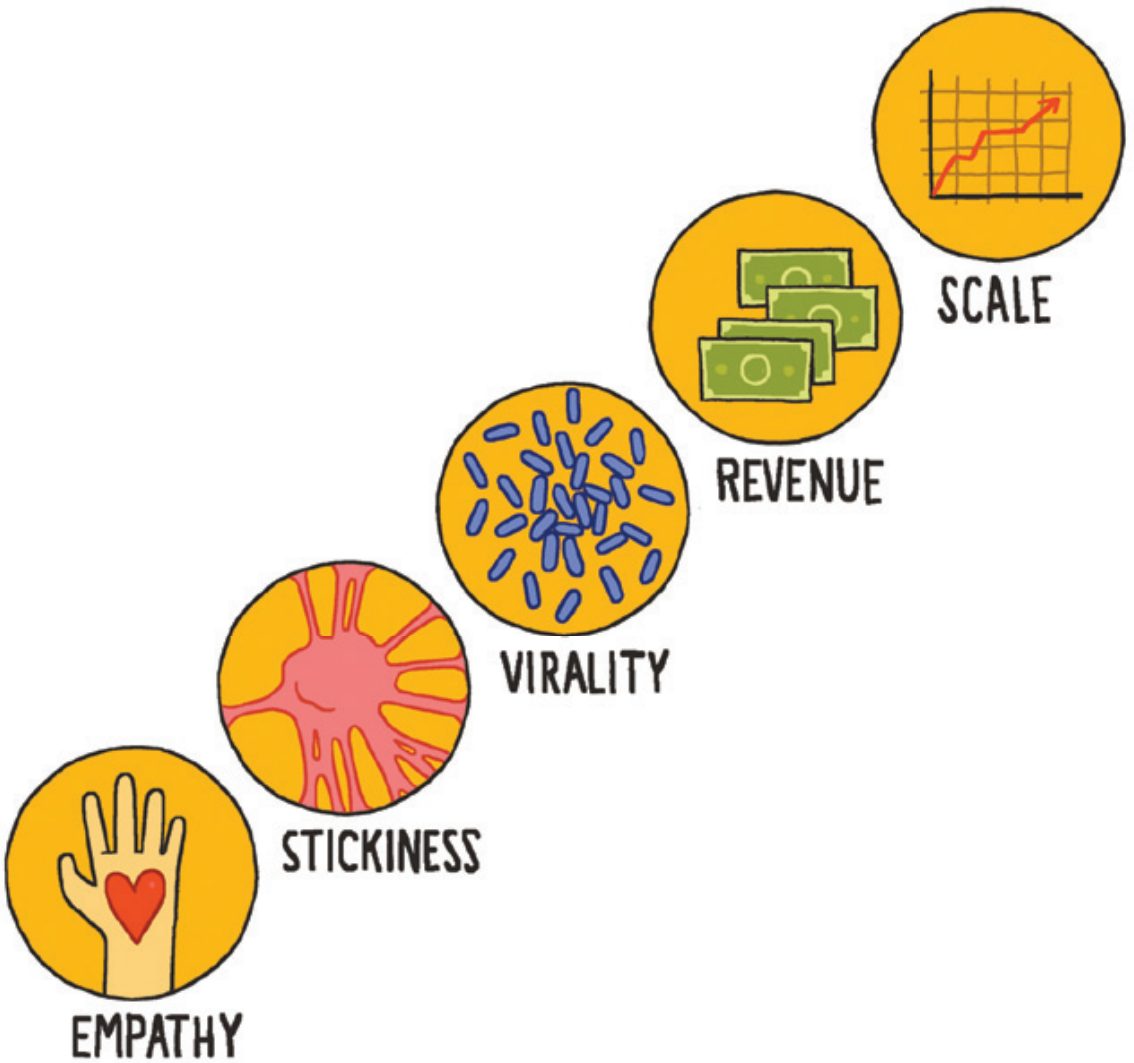
Turn global concerns into design constraints, instead of ignoring predictable problems




FIFTY
Quick
IDEAS

PRIORITISE USING STAGES OF GROWTH

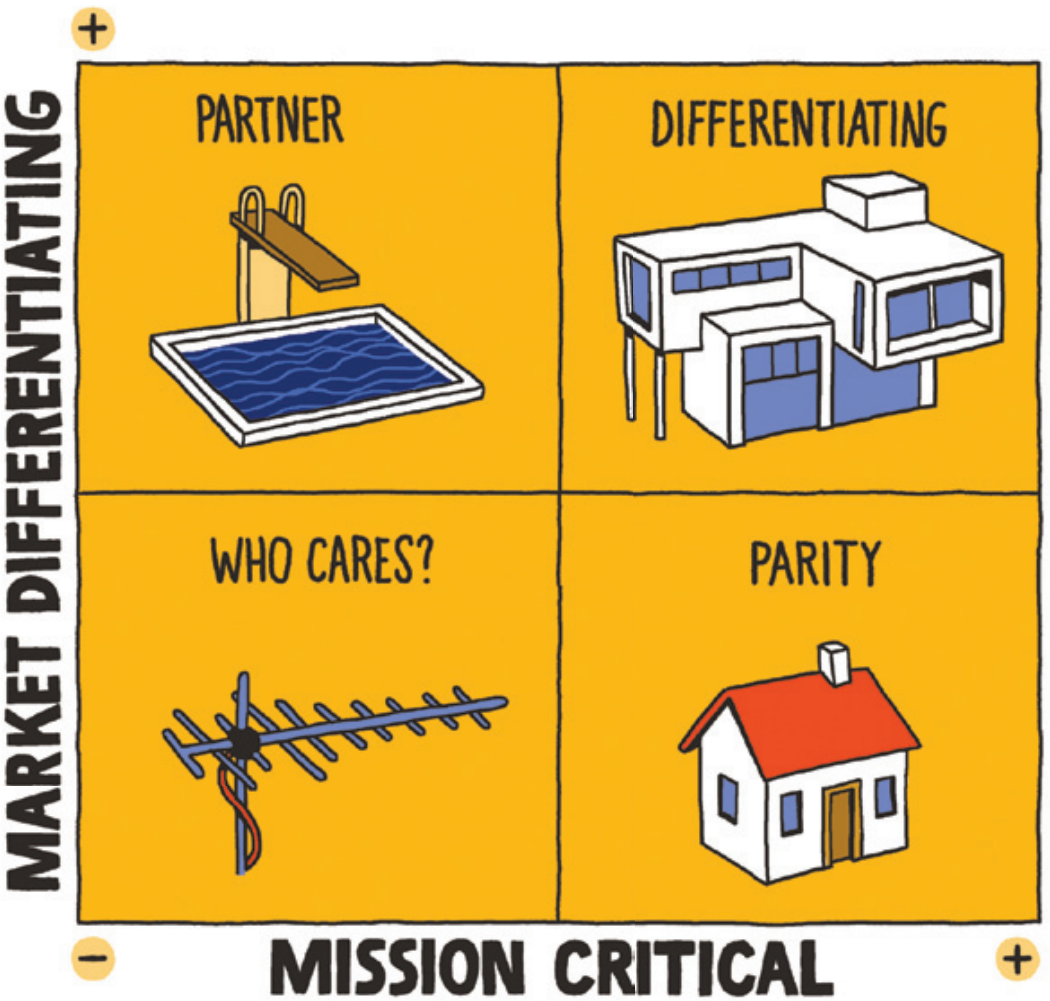
User stories need to be aligned with current business priorities




FIFTY
Quick
IDEAS

PRIORITISE USING PURPOSE ALIGNMENT

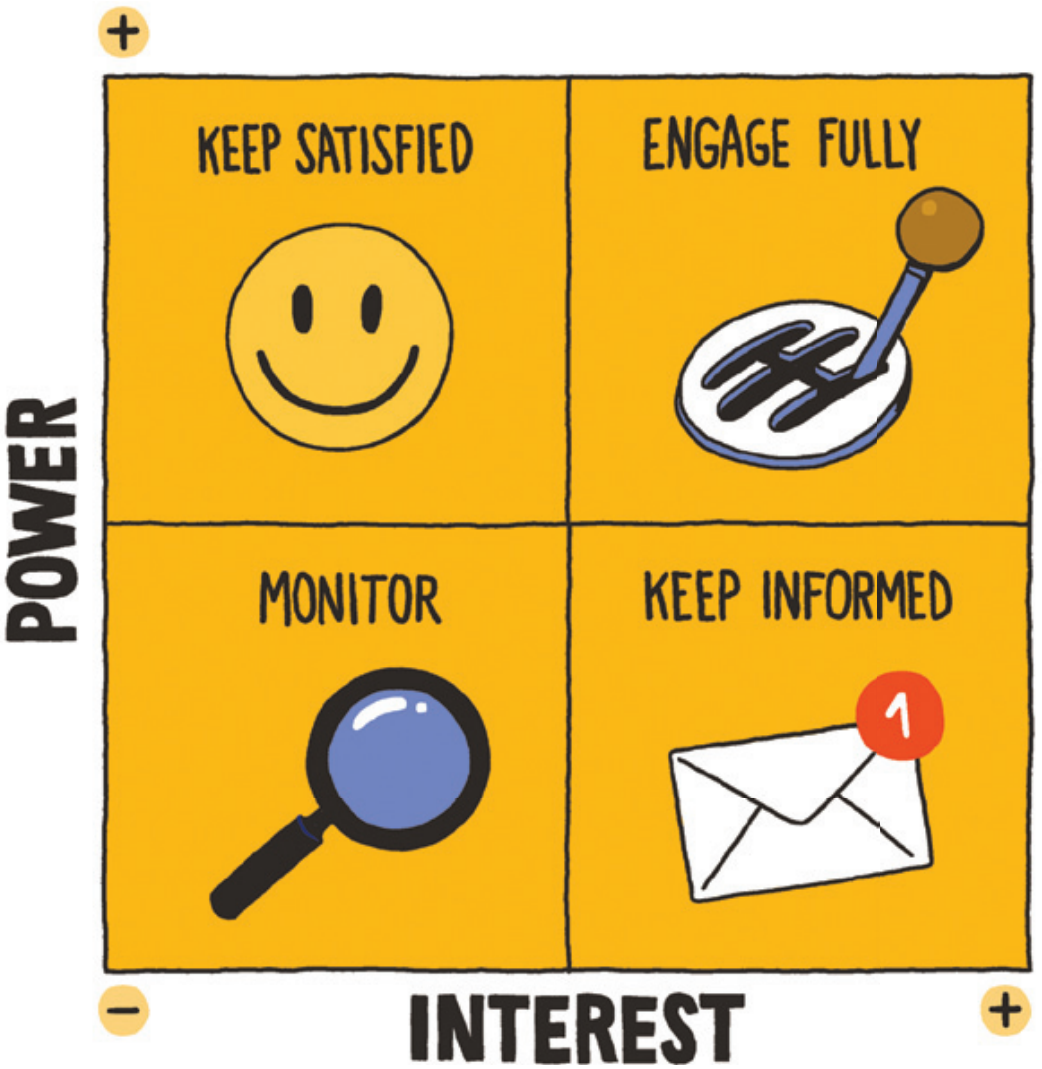
Split stories into those where you need to excel, and those where you just have to be good enough




FIFTY
Quick
IDEAS

MAKE A STAKEHOLDER CHART

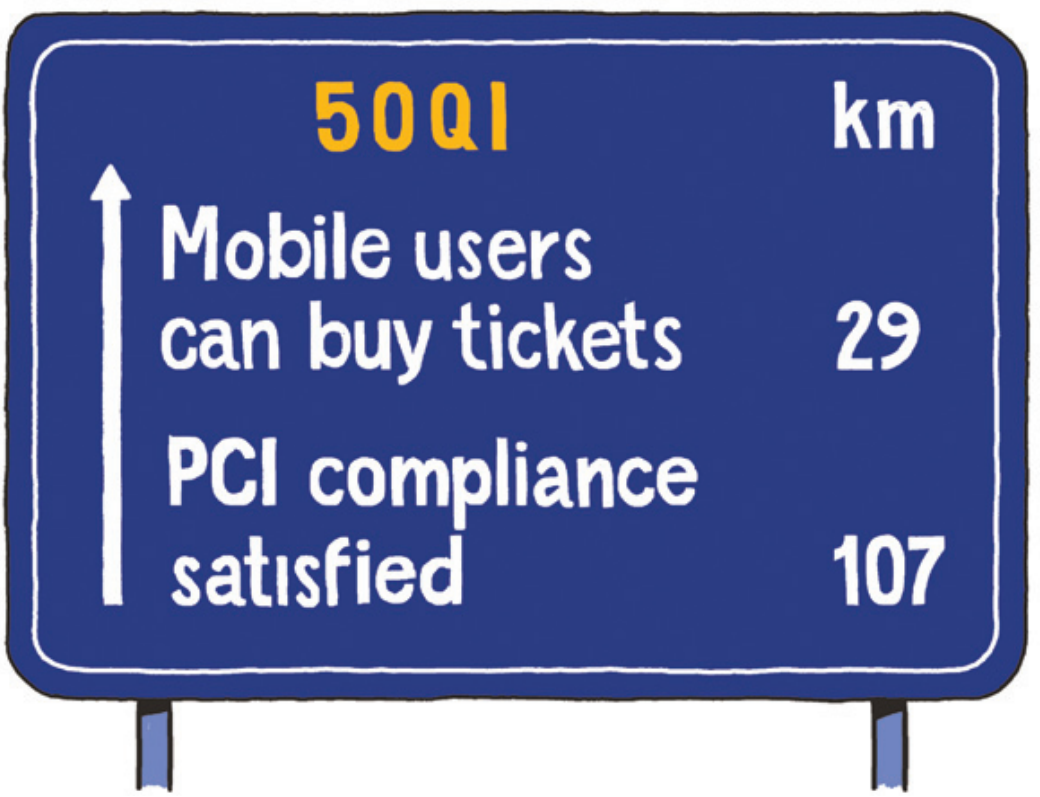
Do not forget big-picture stakeholders who won't necessarily appear in user stories




FIFTY
Quick
IDEAS

NAME YOUR MILESTONES

Meaningful milestone names help stakeholders prioritise




FIFTY
Quick
IDEAS

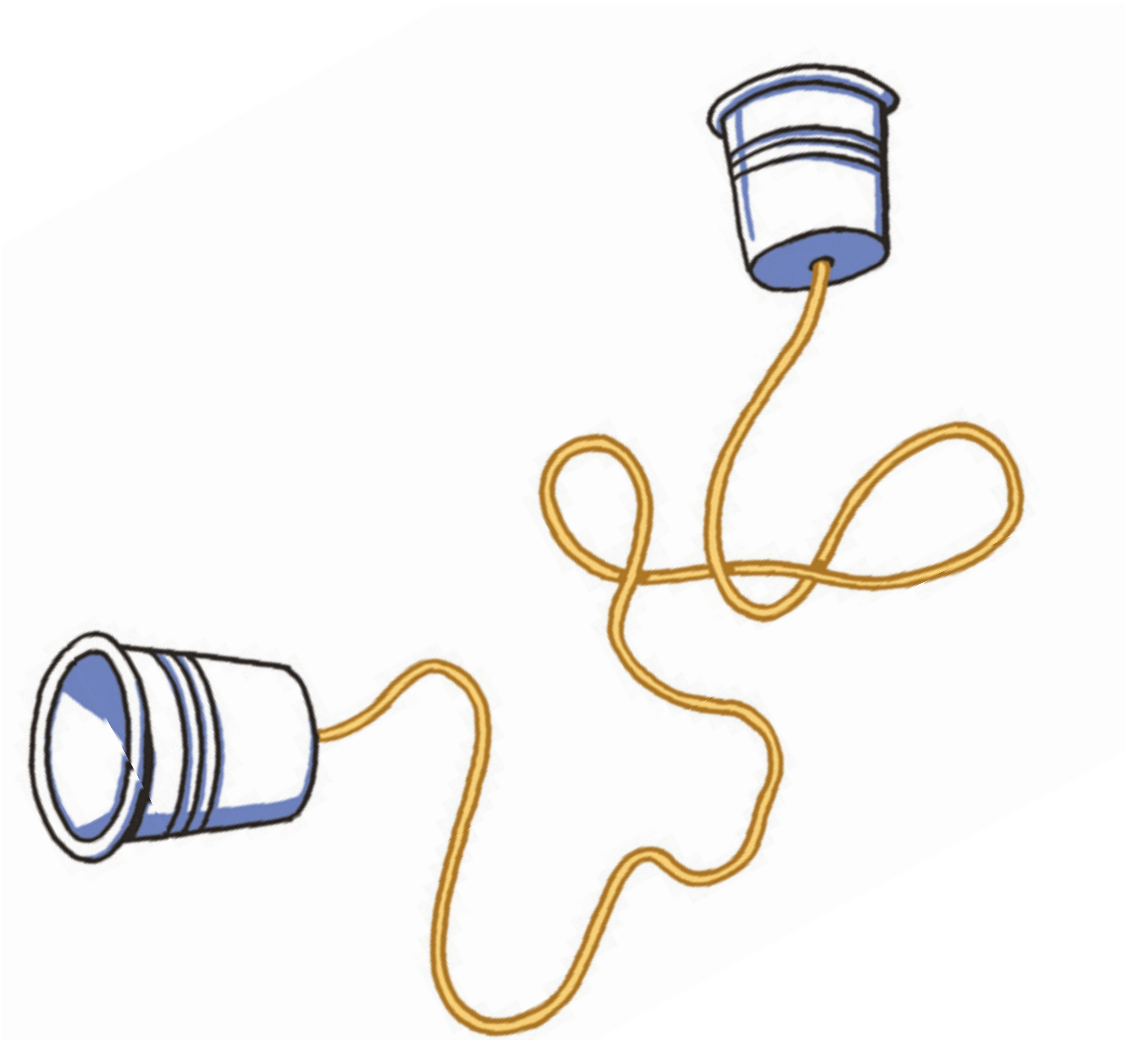
FOCUS MILESTONES ON A LIMITED NUMBER OF USER SEGMENTS

Select target users to prevent generic stories and reduce scope creep


FIFTY
Quick
IDEAS

USE LOW-TECH FOR STORY CONVERSATIONS

Discussions around whiteboards are faster and more productive than using a technical tool




FIFTY
Quick
IDEAS

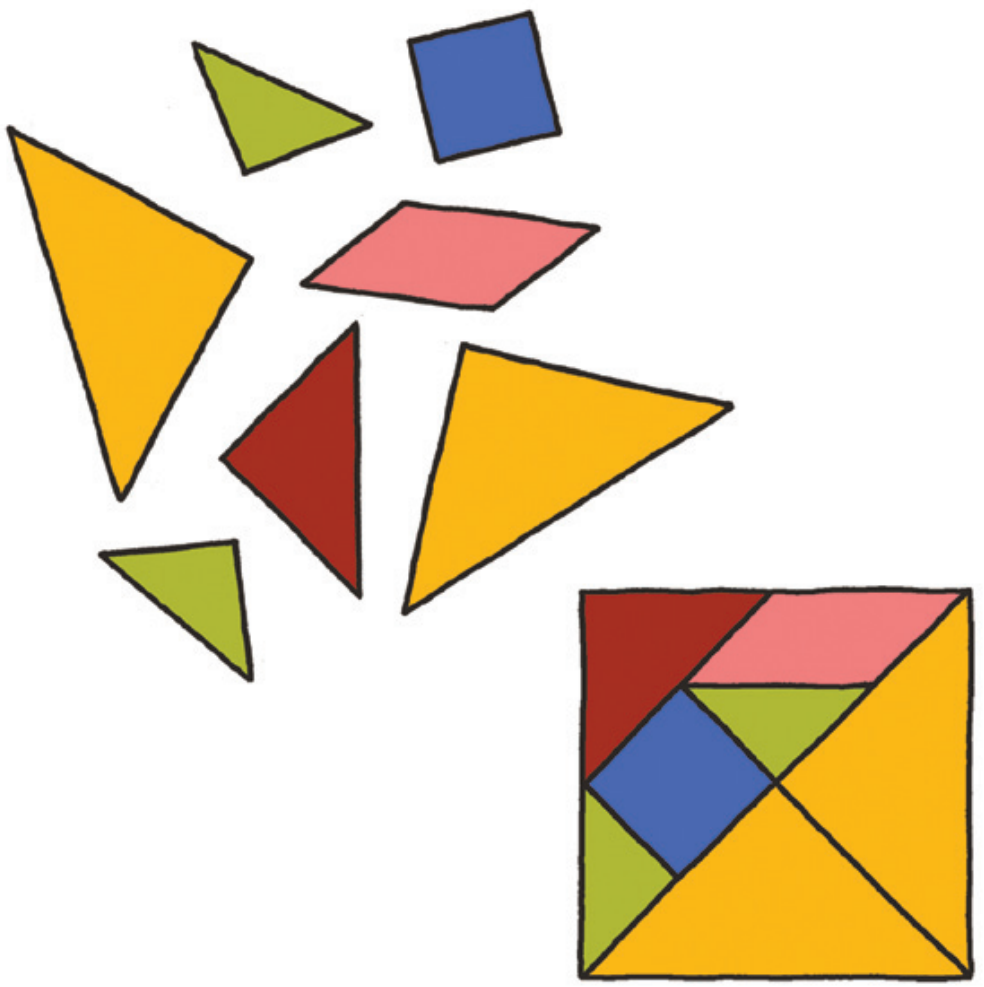
IMAGINE THE DEMONSTRATION

Work out how you will show that you have met the requirements


FIFTY
Quick
IDEAS

DIVERGE AND MERGE FOR STORY DISCUSSIONS

Work in smaller groups to increase the participation of individual team members




FIFTY
Quick
IDEAS

INVOLVE ALL ROLES IN THE DISCUSSION

Instead of delegating analysis to a single person, involve various roles to cover all the perspectives


FIFTY
Quick
IDEAS

MEASURE ALIGNMENT USING FEEDBACK EXERCISES

Measure shared understanding instead of asking ‘Do you have any questions?’


FIFTY
Quick
IDEAS

PLAY THE DEVIL'S ADVOCATE

Challenge expressed user needs and roles to discover fake stories early


FIFTY
Quick
IDEAS

DIVIDE RESPONSIBILITY FOR DEFINING STORIES

Let stakeholders express the need, and those who can design software express the solution


FIFTY
Quick
IDEAS

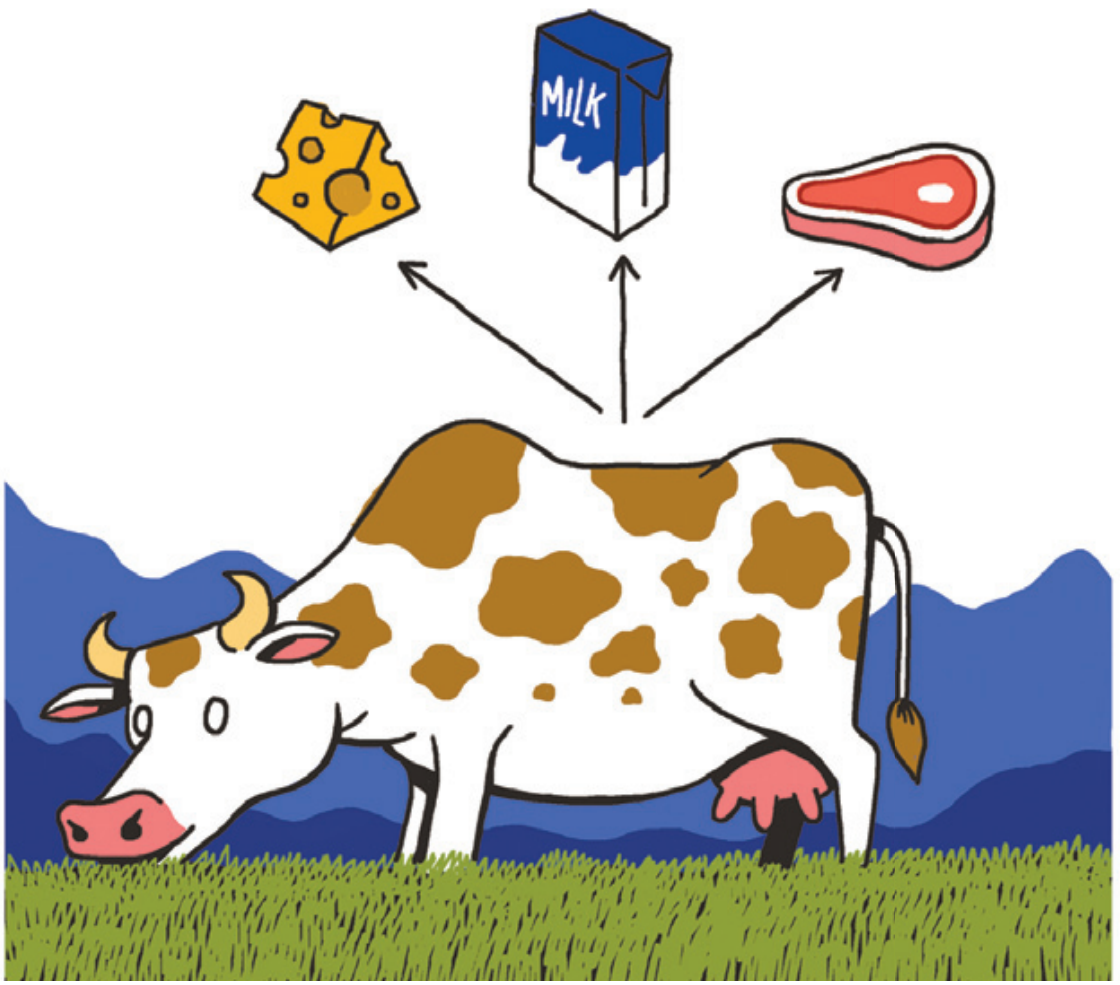
SPLIT BUSINESS AND TECHNICAL DISCUSSIONS

Keep stakeholders interested and engaged by focusing story discussions on their needs


FIFTY
Quick
IDEAS

INVESTIGATE VALUE ON MULTIPLE LEVELS

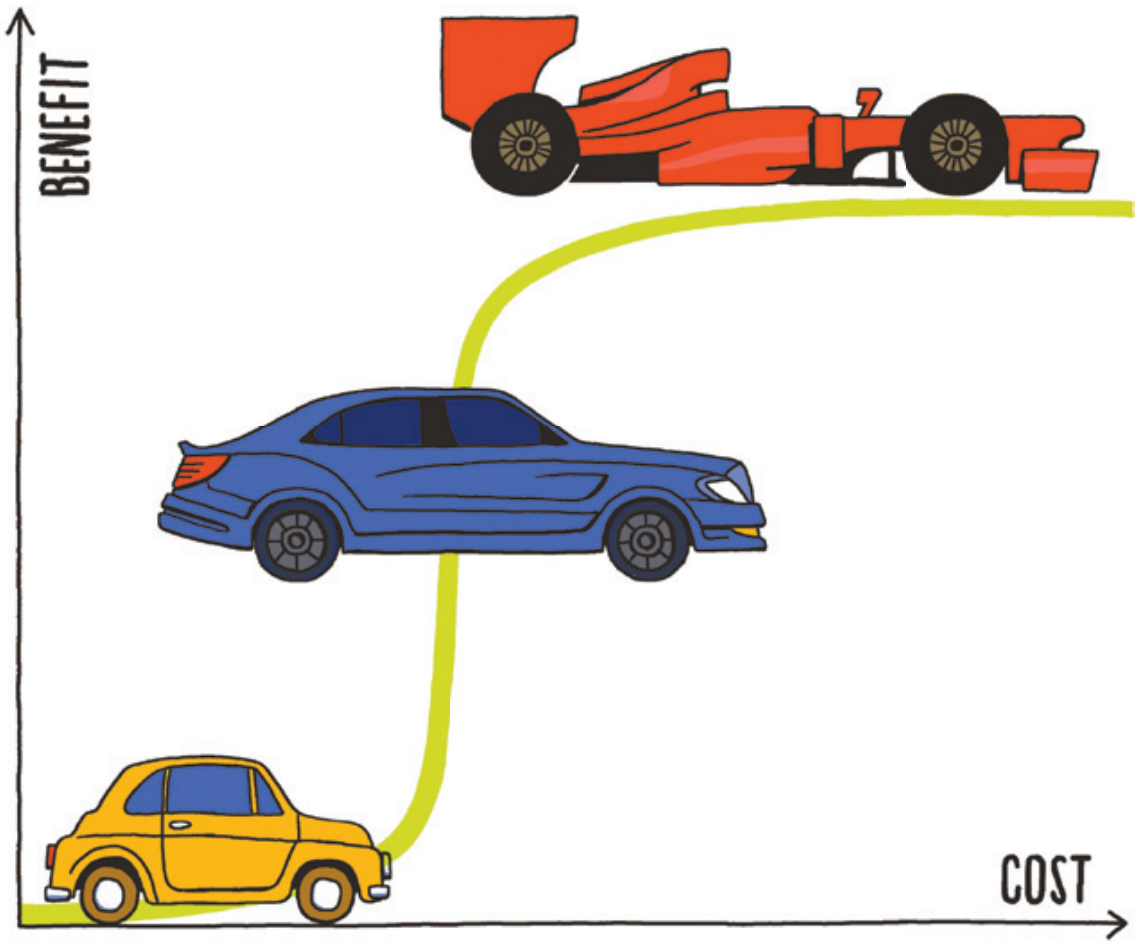
When the whole chain of reasoning is clear, it's much easier to have a useful discussion




FIFTY
Quick
IDEAS

DISCUSS SLIDING-SCALE MEASUREMENTS WITH QUPER

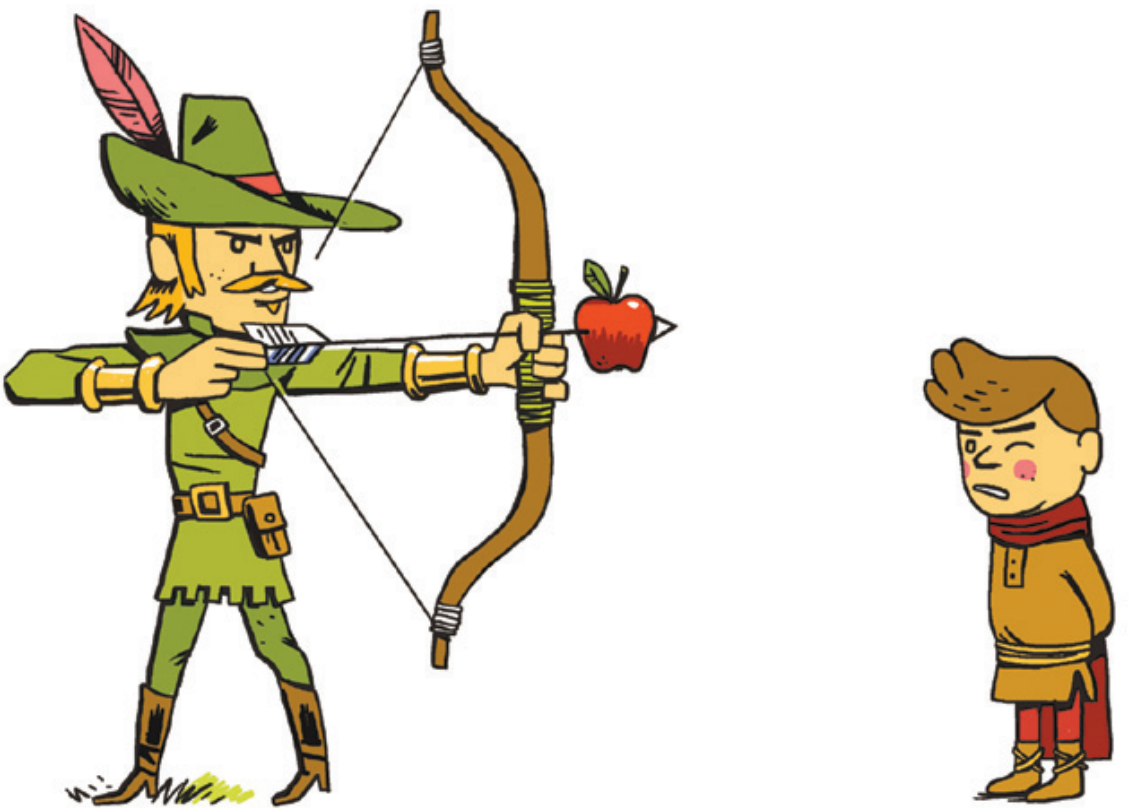
Compare your product to the competition to expose hidden assumptions and requirements




FIFTY
Quick
IDEAS

START WITH THE OUTPUTS

The value of a system is in its outputs, not its inputs




FIFTY
Quick
IDEAS

FORGET THE WALKING SKELETON: PUT IT ON CRUTCHES

Deliver value early by nailing down user interaction, simplify the rest


FIFTY
Quick
IDEAS

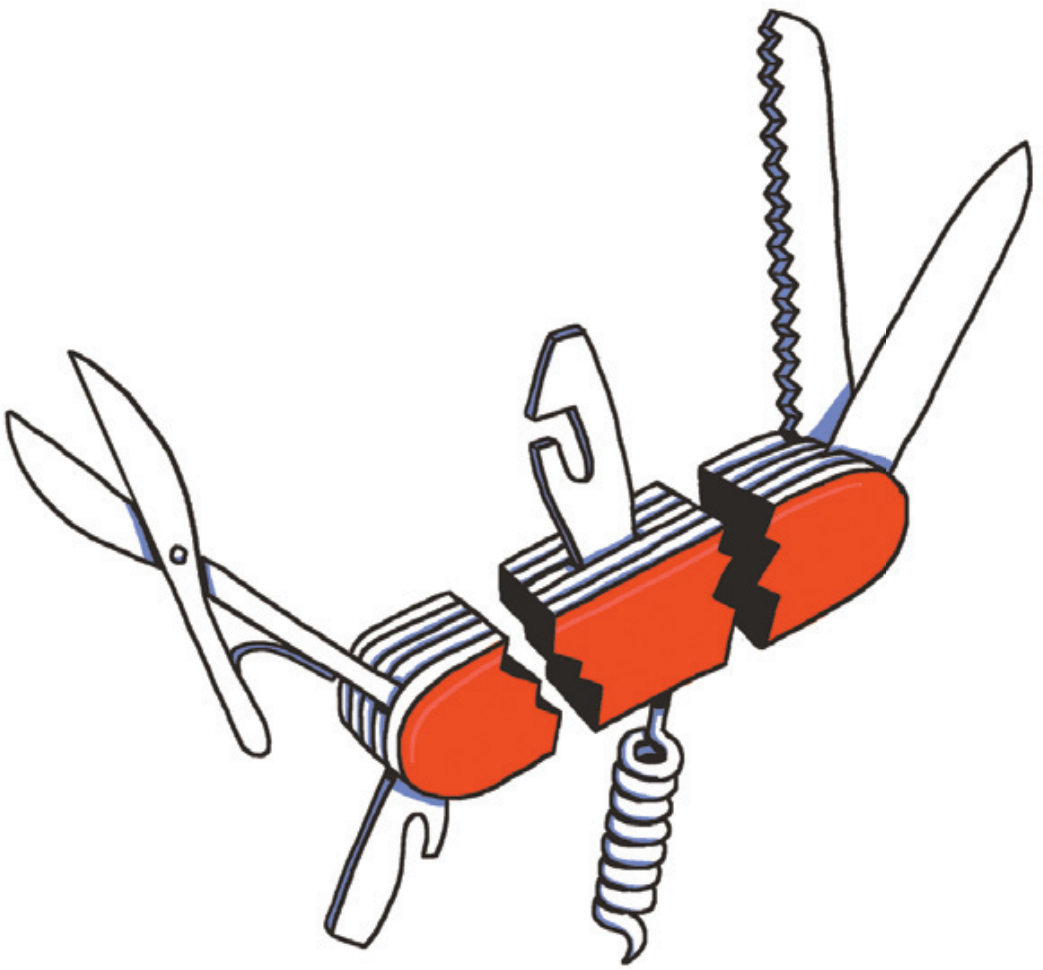
NARROW DOWN THE CUSTOMER SEGMENT

Give 2% of people 100% of what they need, instead of giving 100% of people 2% of needs


FIFTY
Quick
IDEAS

SPLIT BY EXAMPLES OF USEFULNESS

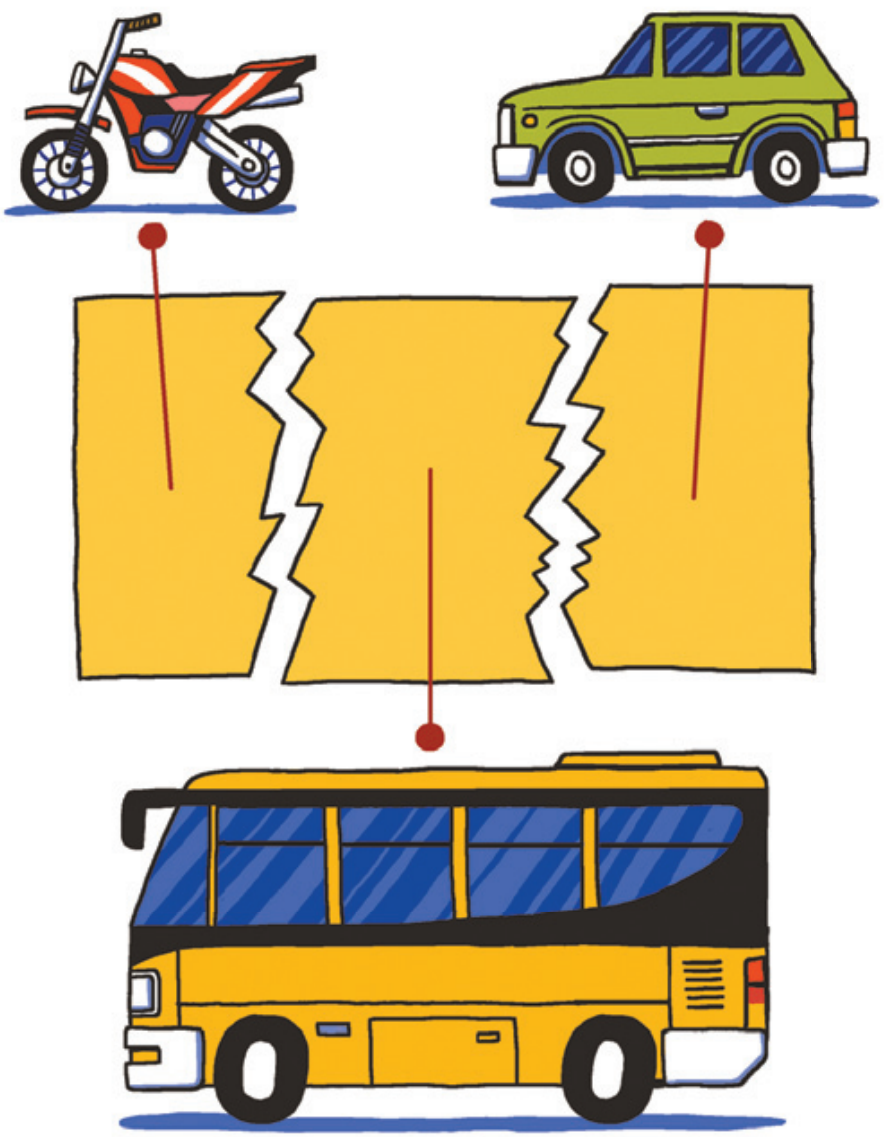
Don’t divide work technically and then look for value – divide by value and look for useful technical chunks




FIFTY
Quick
IDEAS

SPLIT BY CAPACITY

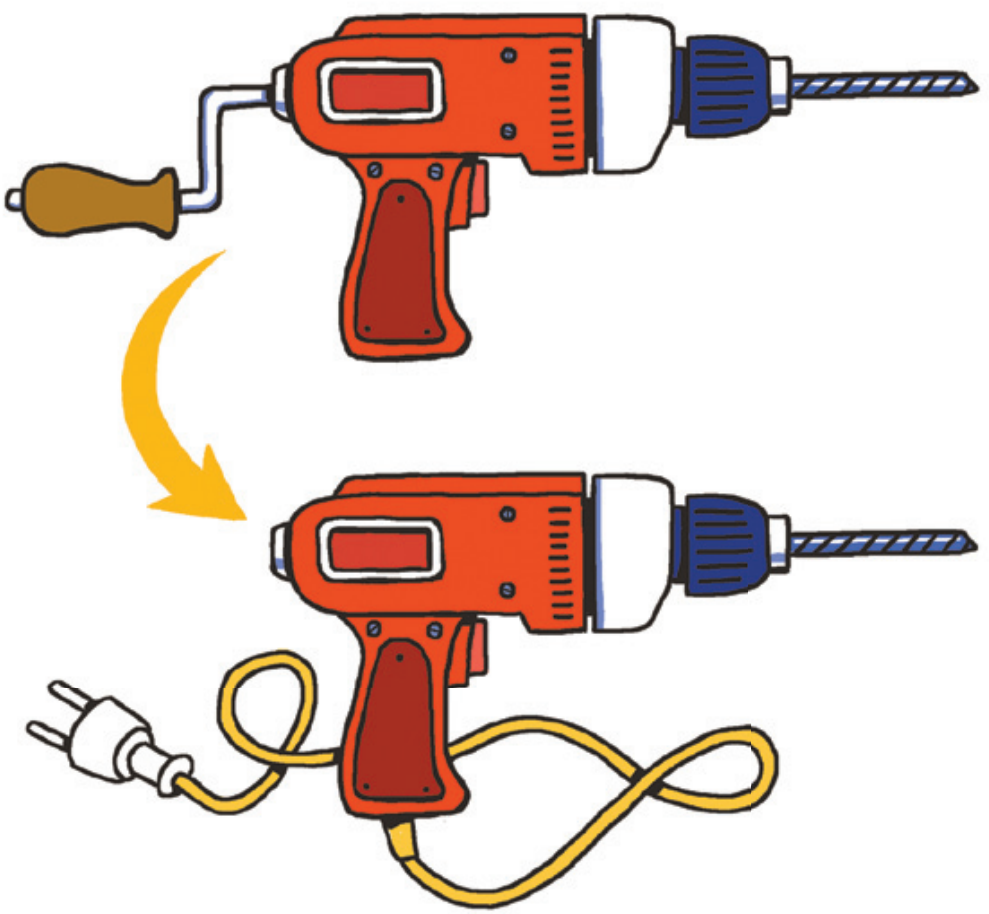
Capacity is often a good way of breaking down ‘all or nothing’ plans




FIFTY
Quick
IDEAS

START WITH DUMMY, THEN MOVE TO DYNAMIC

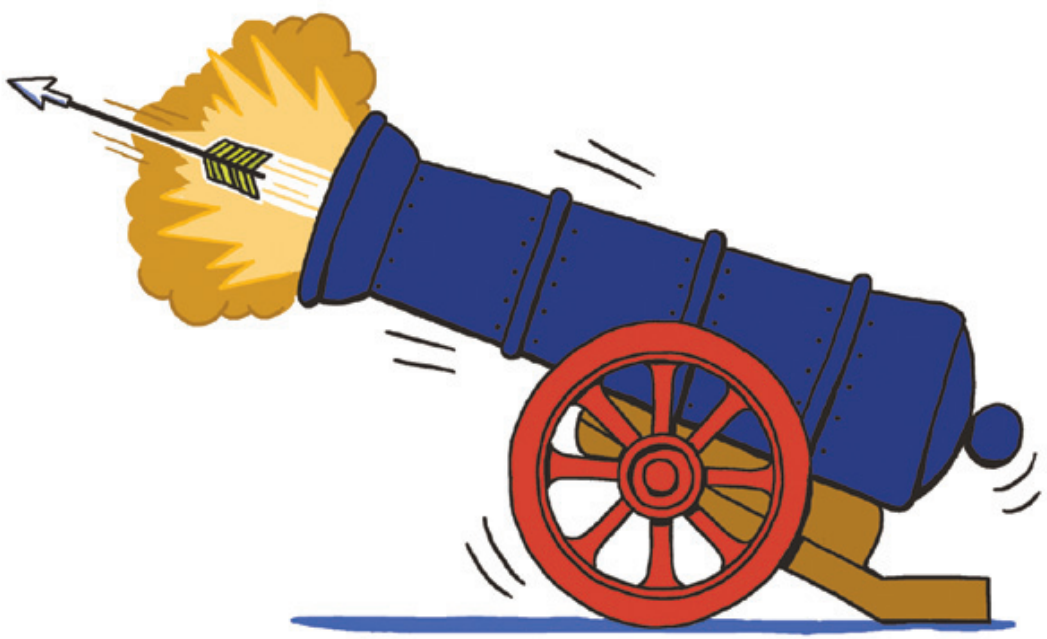
Use hard-coded reference data in first story then connect to databases in a later one




FIFTY
Quick
IDEAS

SIMPLIFY OUTPUTS

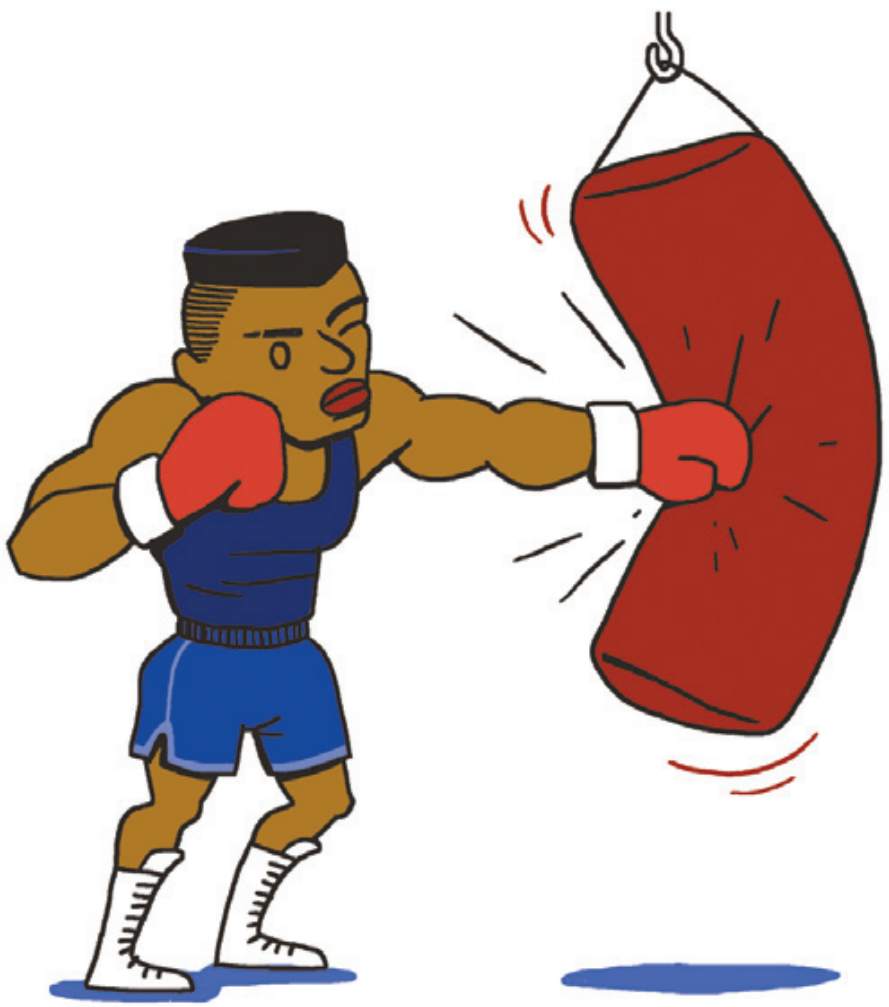
Investigate whether some outputs can be reduced or postponed to de-risk short-term plans




FIFTY
Quick
IDEAS

SPLIT LEARNING FROM EARNING

Time-box research instead of turning it into vague, uncontrolled work




FIFTY
Quick
IDEAS

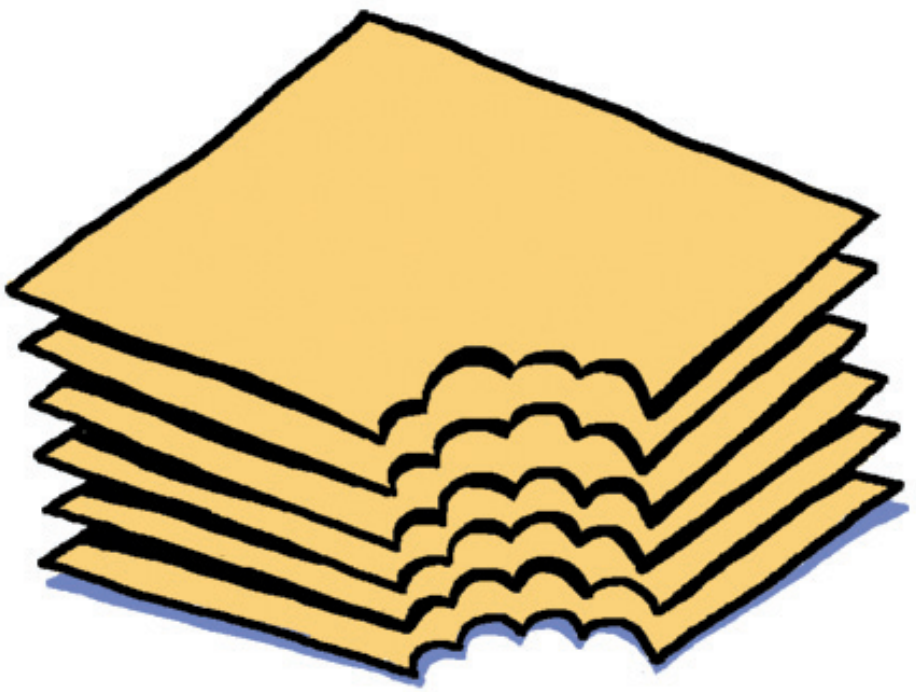
EXTRACT BASIC UTILITY

When business processes cannot be simplified, pare down user interaction to the bare minimum


FIFTY
Quick
IDEAS

WHEN ALL ELSE FAILS, SLICE THE HAMBURGER

Break down technical workflows and group tasks by value




FIFTY
Quick
IDEAS

DON'T PUSH EVERYTHING INTO STORIES

Avoid faking stories to cover internal tasks so that you can keep your plans simple and clean


FIFTY
Quick
IDEAS

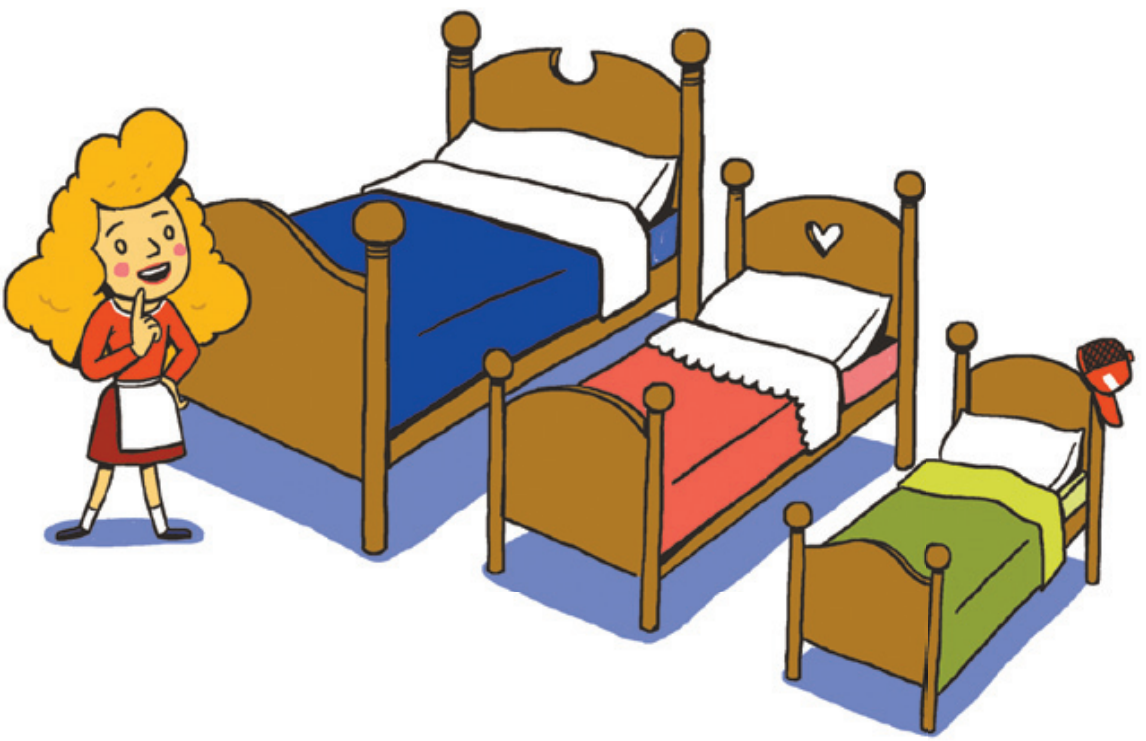
BUDGET INSTEAD OF ESTIMATE

Don’t commit on scope, commit to deliver business value


FIFTY
Quick
IDEAS

AVOID USING NUMERIC STORY SIZES

Don’t add up story sizes – that way people won’t be able to misuse them




FIFTY
Quick
IDEAS

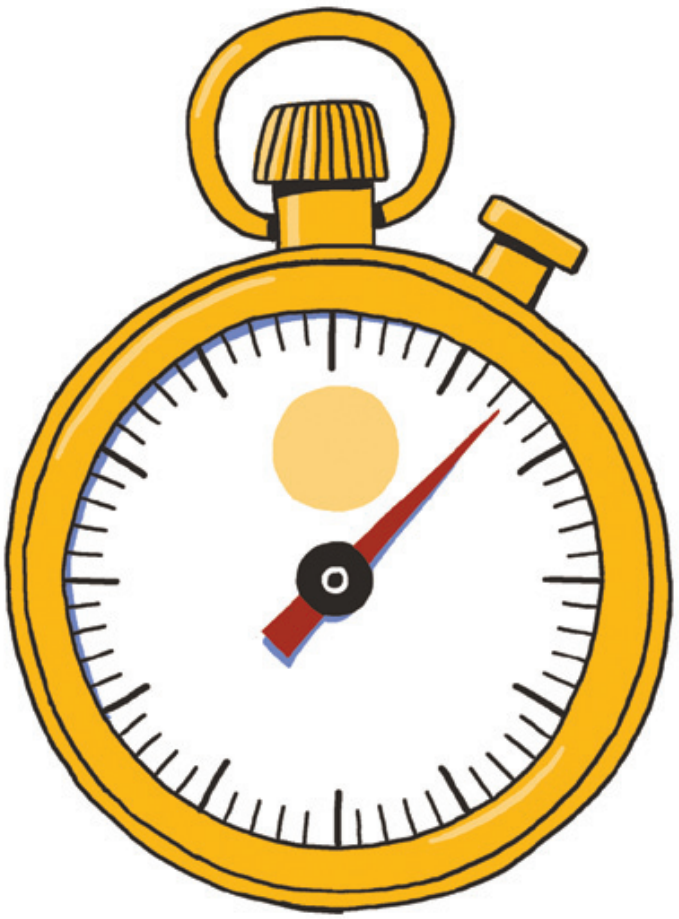
ESTIMATE CAPACITY BASED ON ROLLING NUMBER OF STORIES

If you have to add up story sizes, use only similar stories so that averages make sense


FIFTY
Quick
IDEAS

ESTIMATE CAPACITY BASED ON ANALYSIS TIME

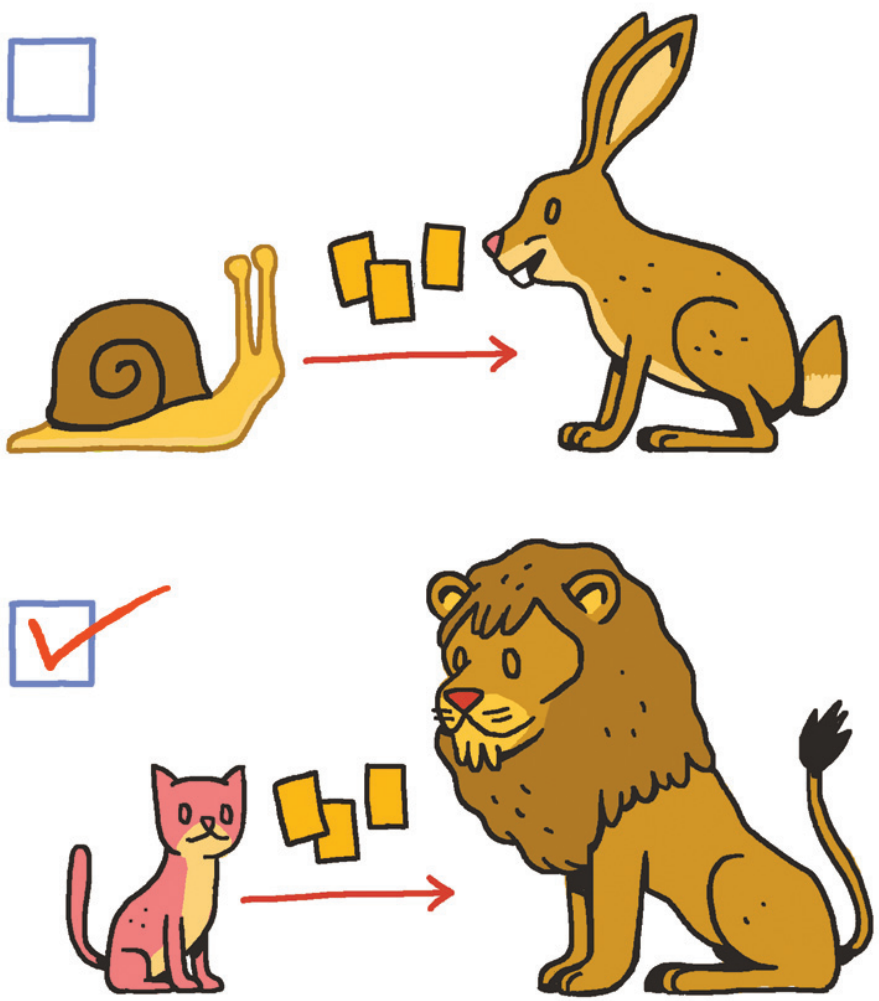
If you don't have time to discuss a story, you probably won't have time to deliver it




FIFTY
Quick
IDEAS

PICK IMPACTS INSTEAD OF PRIORITISING STORIES

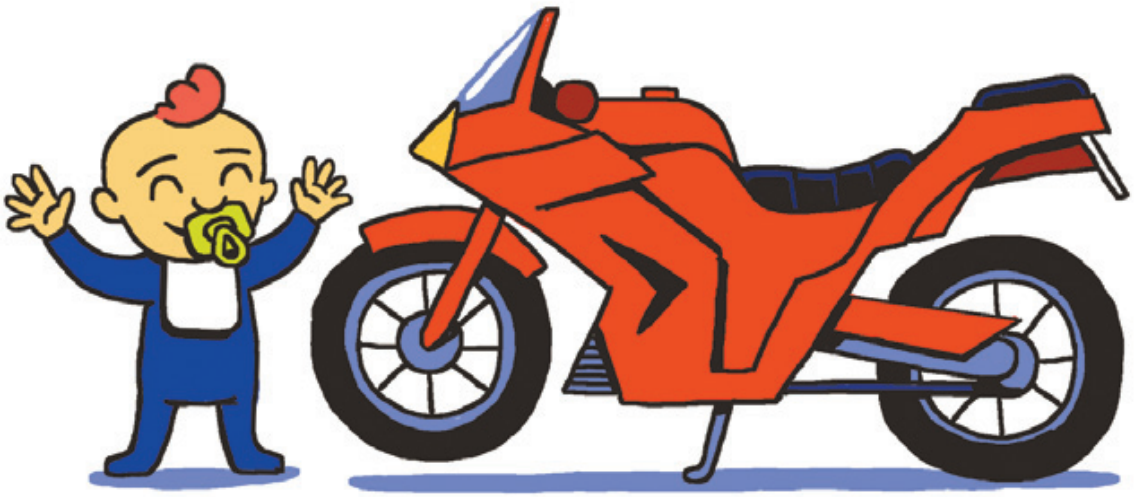
Focus on achieving a big impact instead of balancing work so that nobody gets angry




FIFTY
Quick
IDEAS

NEVER SAY 'NO' – SAY 'NOT NOW'

Negotiate sequence of delivery instead of refusing requests




FIFTY
Quick
IDEAS

SPLIT UX IMPROVEMENTS FROM CONSISTENCY WORK

Manage consistency work and large interaction improvements separately


FIFTY
Quick
IDEAS

GET USERS TO OPT IN TO LARGE USER INTERFACE CHANGES

Roll out UX changes gradually by offering something important and asking people to opt in


FIFTY
Quick
IDEAS

CHECK OUTCOMES WITH REAL USERS

Check with real users whether they actually got expected benefits


FIFTY
Quick
IDEAS

THROW STORIES AWAY AFTER THEY ARE DELIVERED

Specifications and tests should explain how a system works currently – not how it changed over time