QUALITY INTELLIGENCE

Intelligence about software quality, intelligently applied

Some Models Are Useful

Using Models in Your Software Test Strategies

Fiona Charles

TiD 2018

A test strategy is a set of big-picture ideas and decisions embodying the overarching design of a test or test campaign

Your solution to the problem

How to uncover the most important information about the system

Most efficiently & effectively

Within the constraints

With the resources available to you

While managing the risks to your testing

Information that matters conveys something significant about system value:

Confirmation of documented requirements

Bugs that impair, impede or threaten value

“Quality is value to some person or persons.”

Gerald M. Weinberg

Value(s)

Business drivers for the project

Benefits that stakeholders expect this project to deliver with/in this system

The design behind the plan

The core of any test strategy is the model

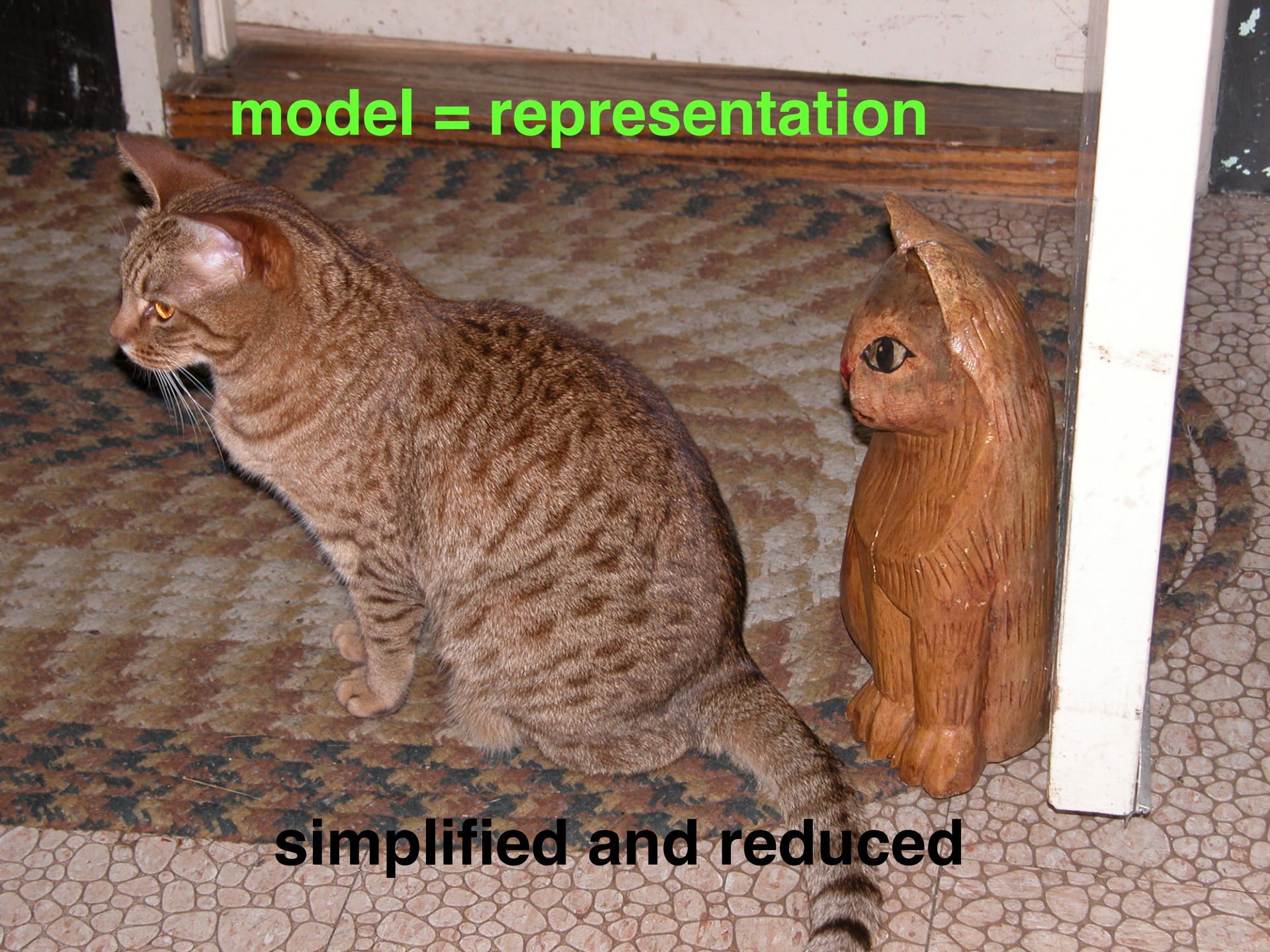
That's true even if:

Your test is driven entirely by user stories

Or BDD statements

Or use cases

Or requirements…


model = representation
simplified and reduced

"Every model is ultimately the expression of one thing…we hope to understand in terms of another that we do understand."

Gerald M. Weinberg, An Introduction to General Systems Thinking

test model

a simplified and reduced representation of the feature, system or set of business processes we are testing

The model we choose gives form to our strategic choices

Why model consciously?

In every test, we make choices

- What to include and what to leave out
- What to base the test on
- How to test

Consciously modeling a test gives us a way to control, examine, and explain those choices

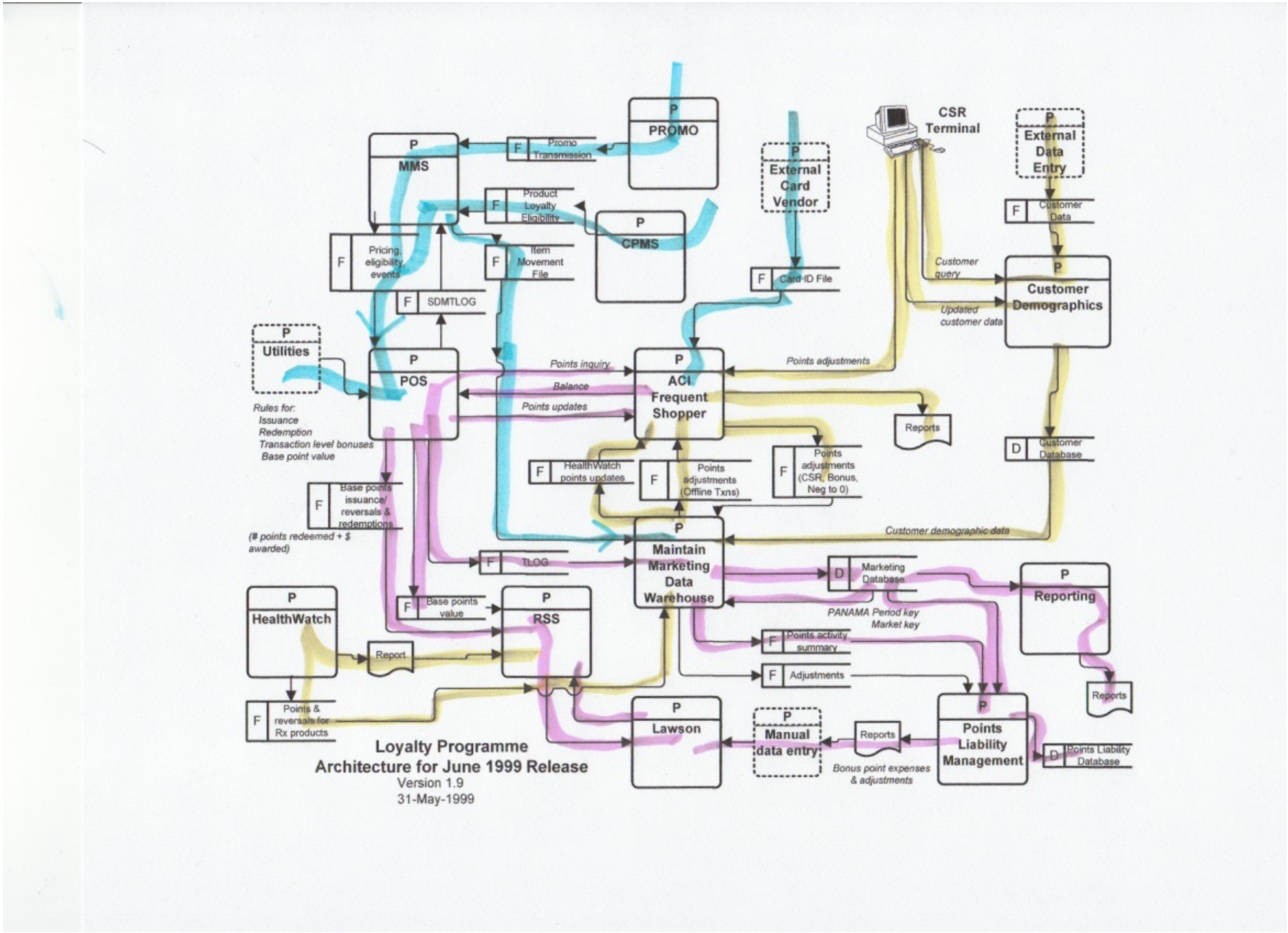
A useful model:

Simulates something real about the system or about how the system will be used

Departs from the model the programmers used to create the system

Stimulates interesting test ideas

Is easy for other people to understand



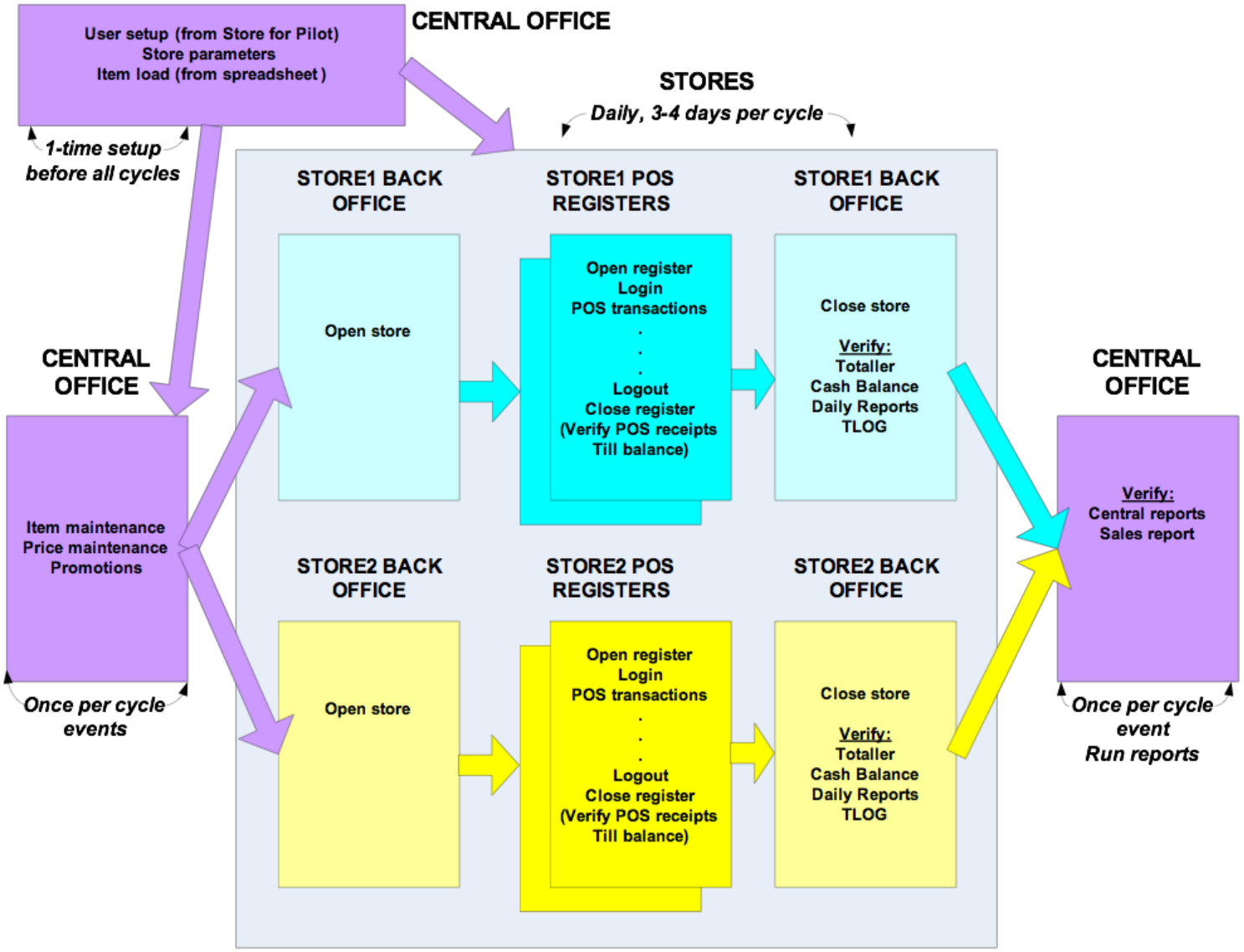
There are many possibilities for designing the model that will best help solve your testing problem

Business operations

Business operations through a retail day, week, month, year

- Product definition & pricing
- Periodic promotions setup and aging
- Open store day
- Sell/return, other transactions
- Close store day
- Balance the day
- Produce financial reports
- Multiple stores + Central office

Model for testing a POS system



Functional or organizational decomposition

Functional areas within the system or business (Ordering, Inventory Management, Billing, etc.)

Processes in each area (Order capture, provisioning, etc.)

Functions within each process (Enter, edit, cancel order, etc.)

System (or integration) data

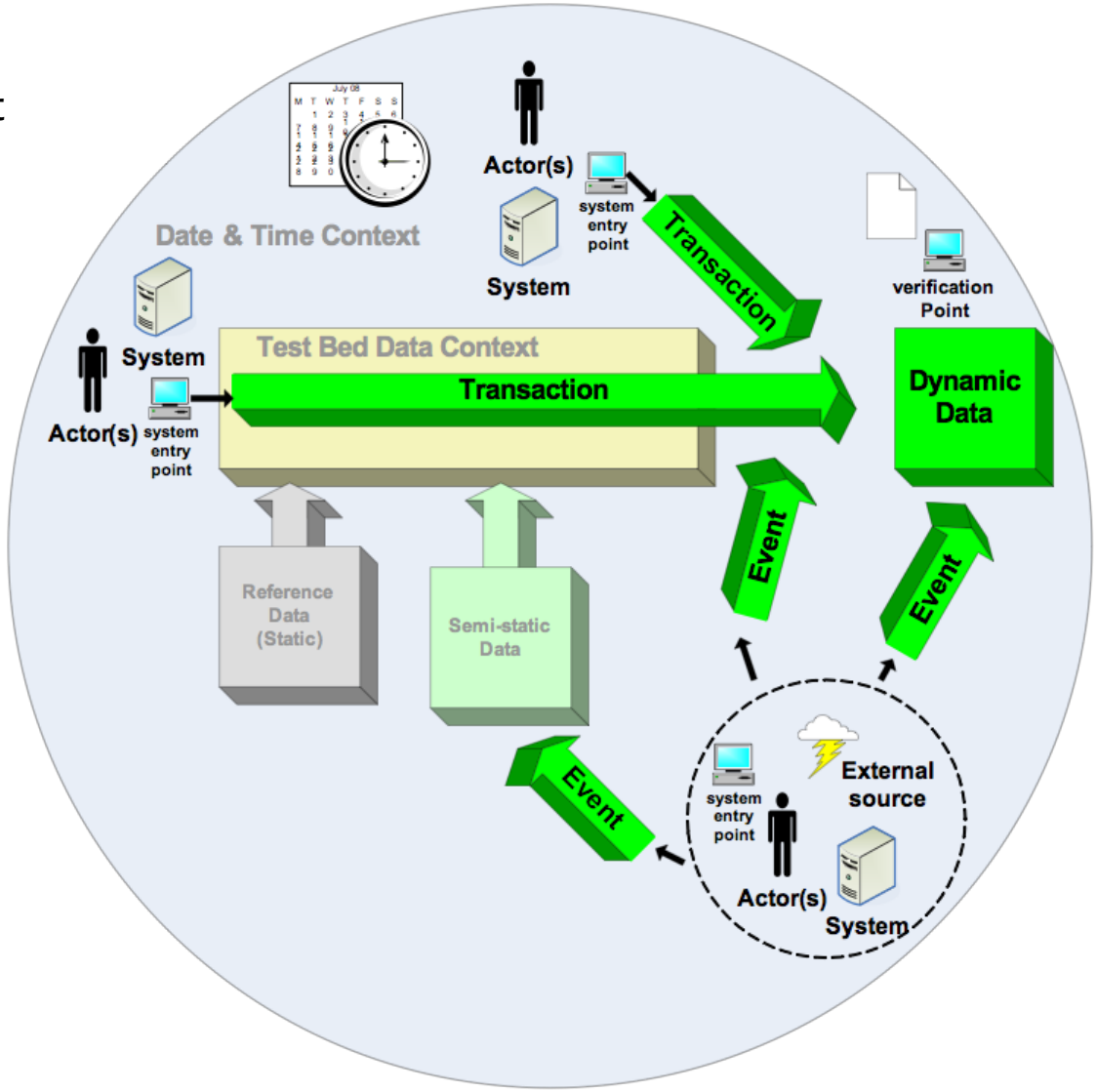
Static

Semi-static

Dynamic

(see my *Modelling Scenarios using Data*)

A high-level framework based on data can be useful for modeling a test of a transactional system (or multi-system solution).



A data-driven model

Represents a systems view that can be rounded out using models based on a business view

Identifies principal components of the execution model for the test:

- Setup data
- Entry points
- Verification points
- Events to schedule

Easy to structure and analyze

Lifecycle of any significant entity

A bank account, from open through to close

A purchase, from order through extended warranty servicing

Student lifecycle

Customer lifecycle

Stakeholder experience

Users

Customers

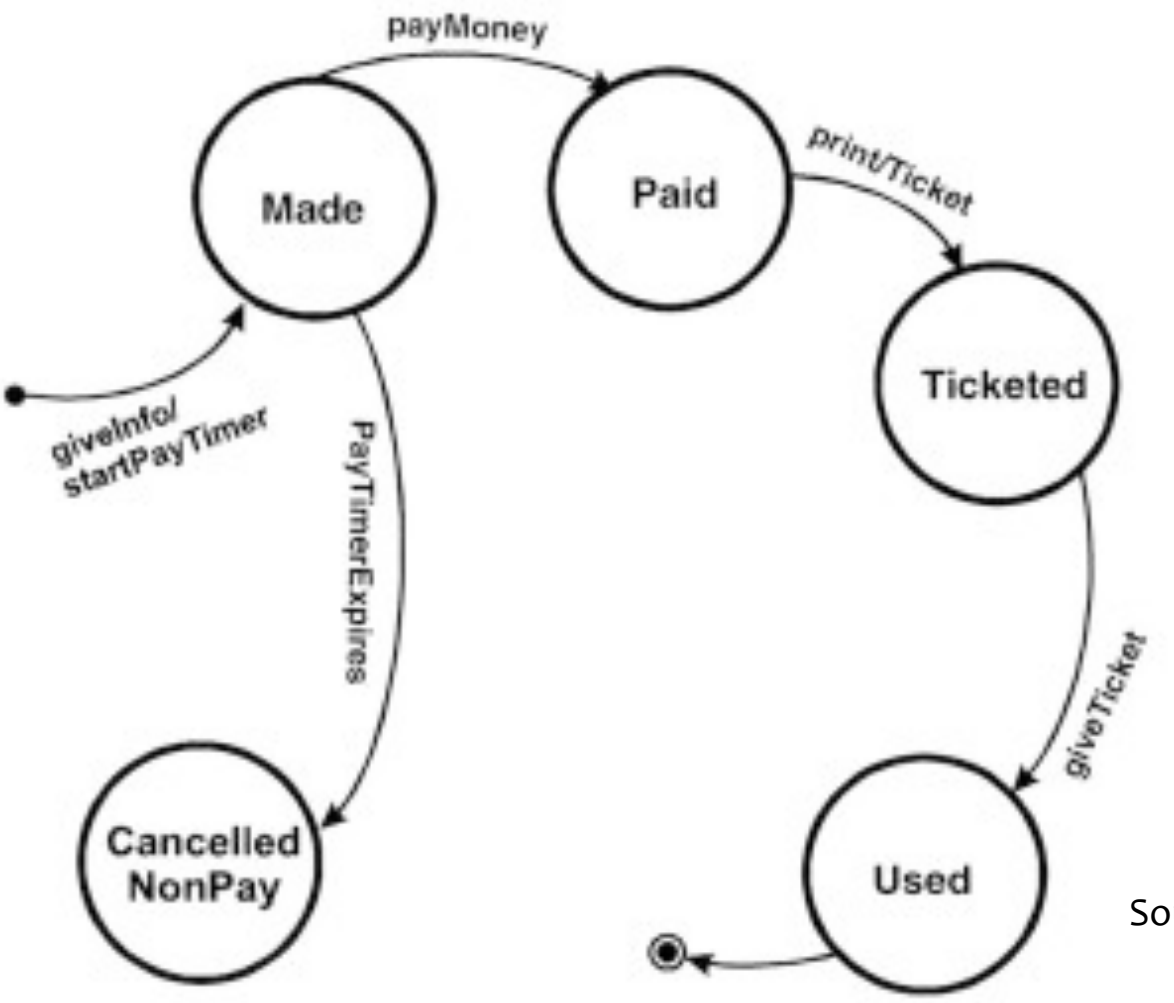
Managers/approvers

Installers

Personae

Familiarity and comfort with technology
Patience/attention span
Motivation
Demographics
Accessibility needs
Locations…

State transitions…



Source: testitquickly.com

It’s a good idea to consider using more than one model

Because no model is “right”

Each model type can act as a source of test ideas

Different models can act as cross-checks on each other, generating ideas for complementary scenarios, e.g.,

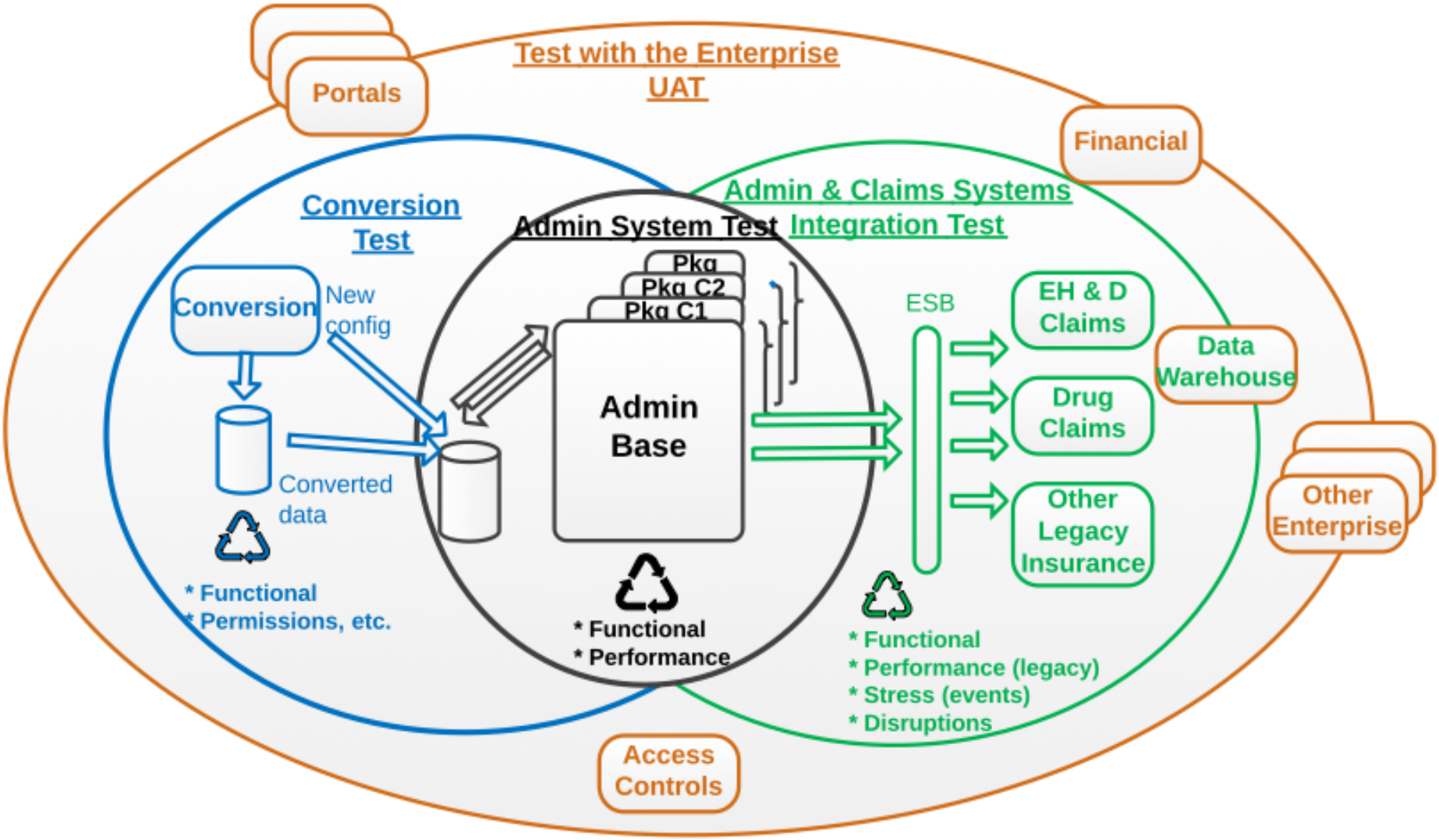
Combining the real-world view of a business operations model with the systems view of a model based on the data

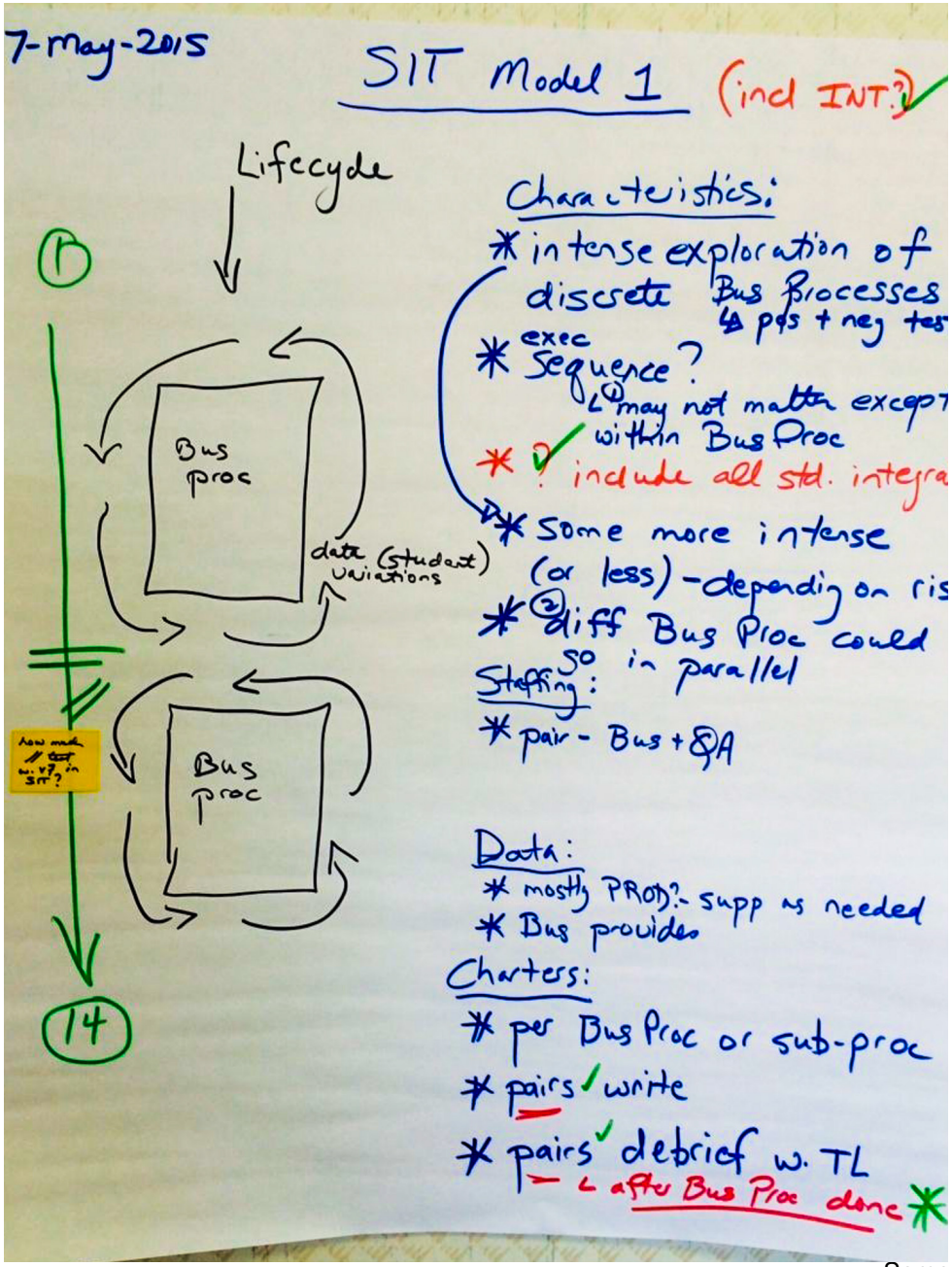
“If it is fish you’re looking for, why climb trees?”

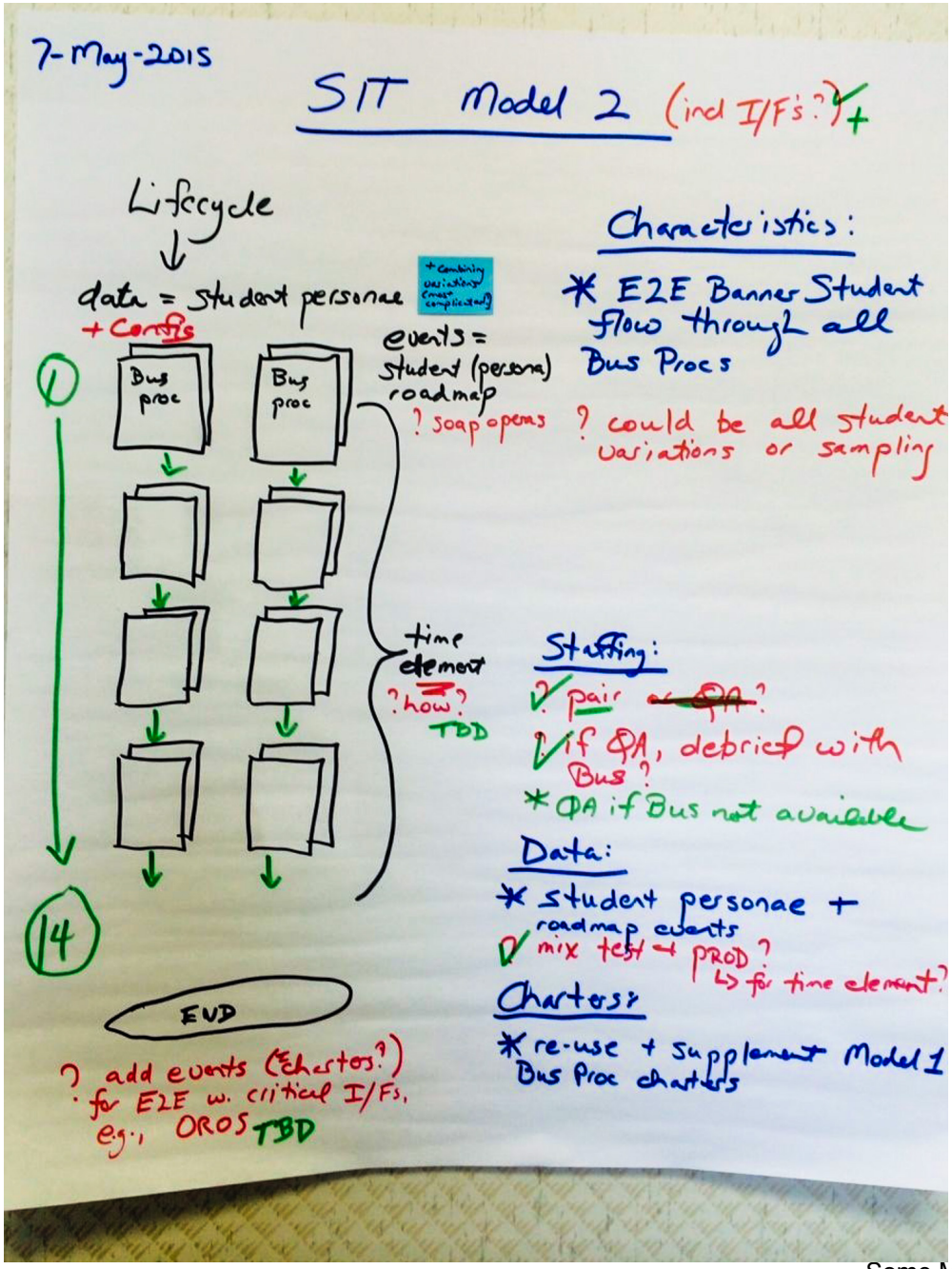
Asian proverb

Modeling at different levels

Overall model for testing on a large insurance project:





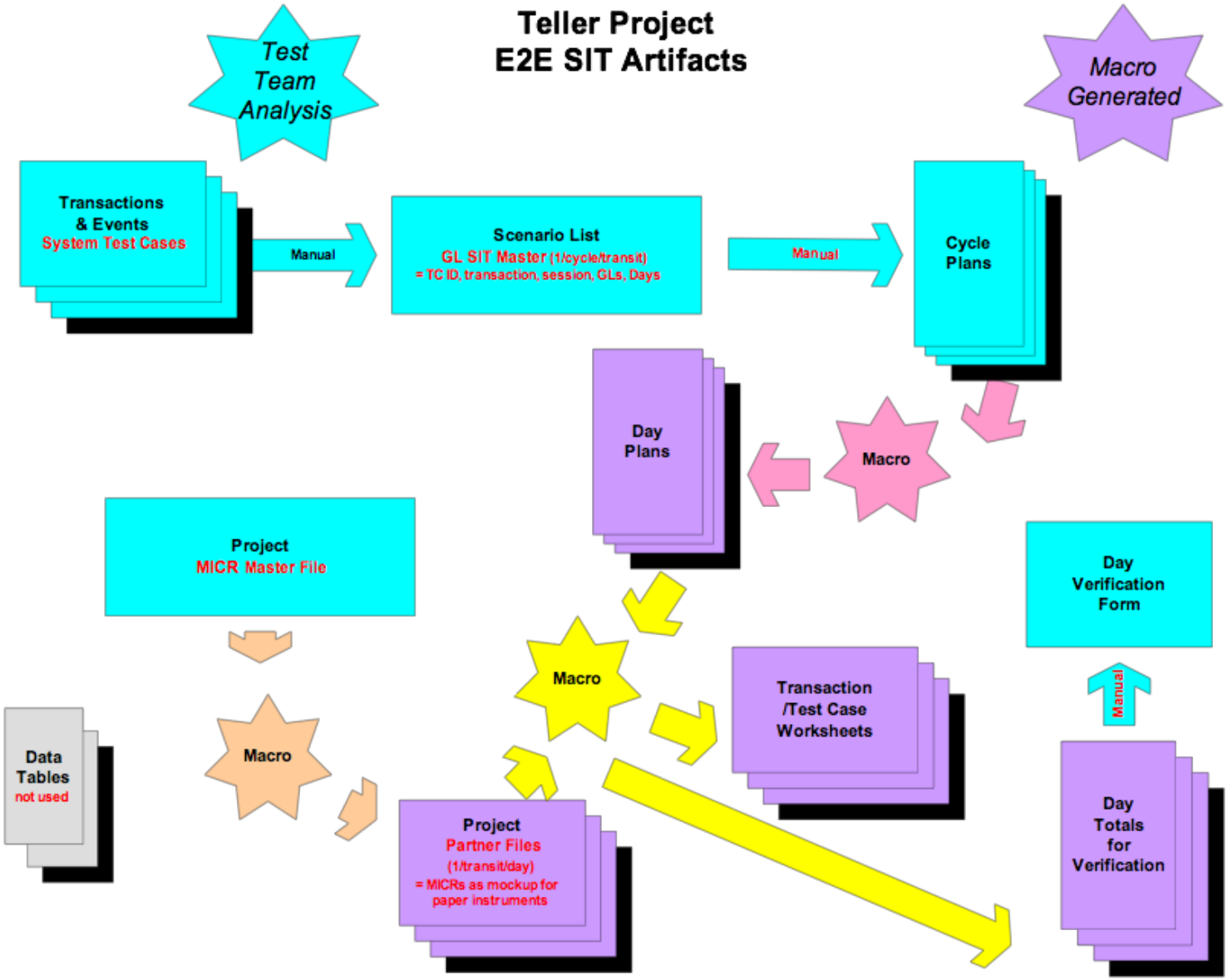


You may need to model more specific strategies

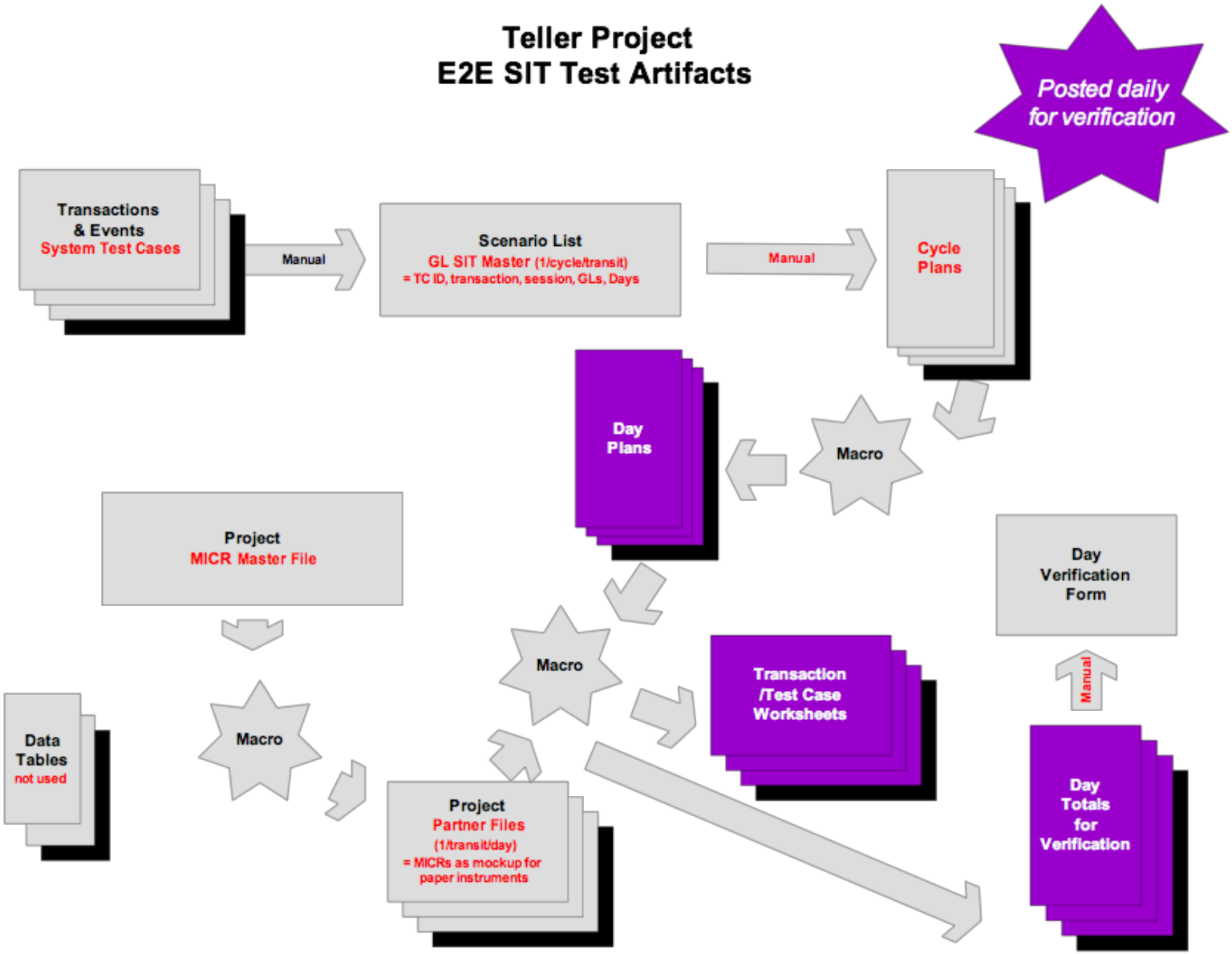
Risk management

Resource management (data, environments, staffing, tools…)

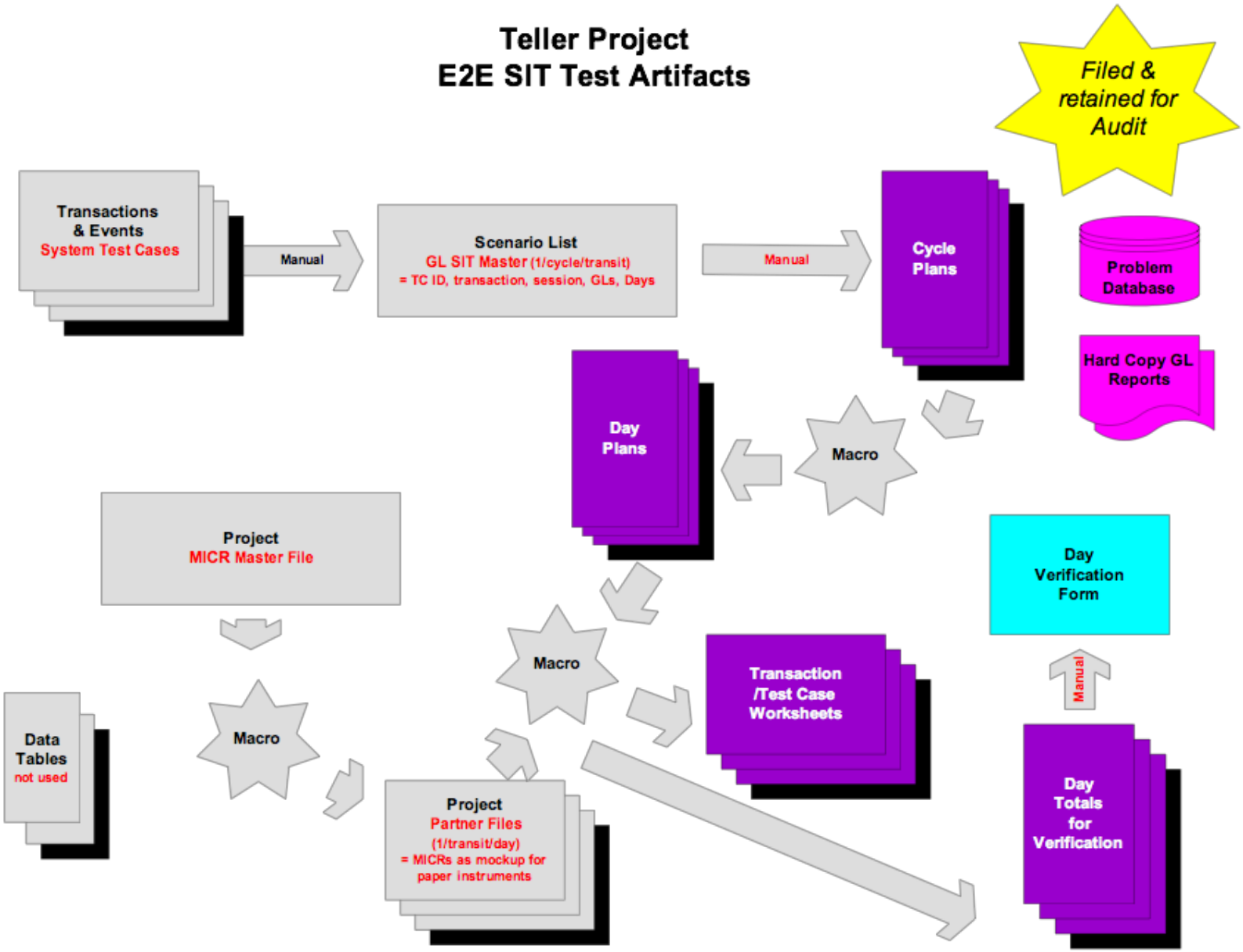
Artifacts



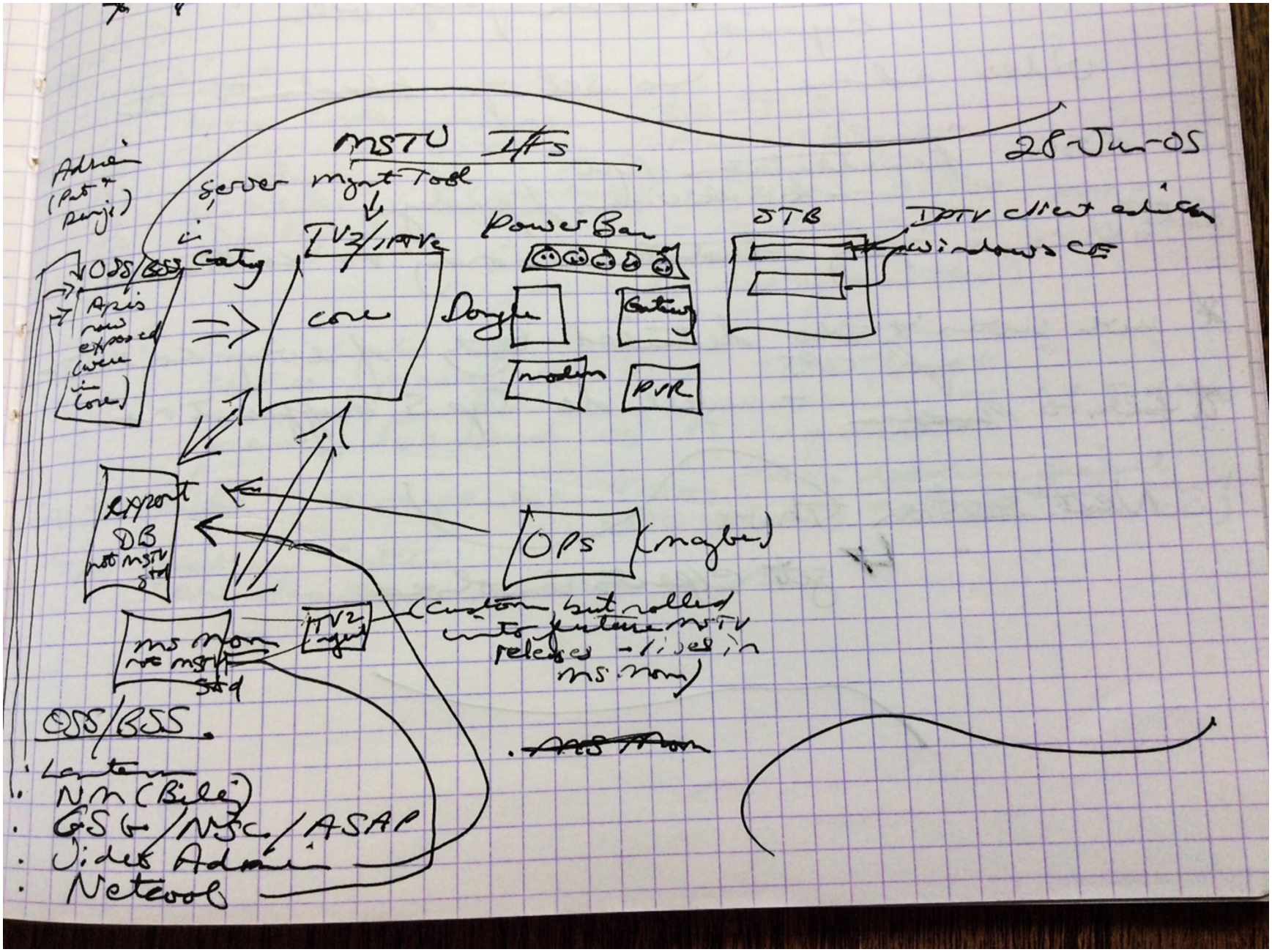
Teller Project
E2E SIT Test Artifacts

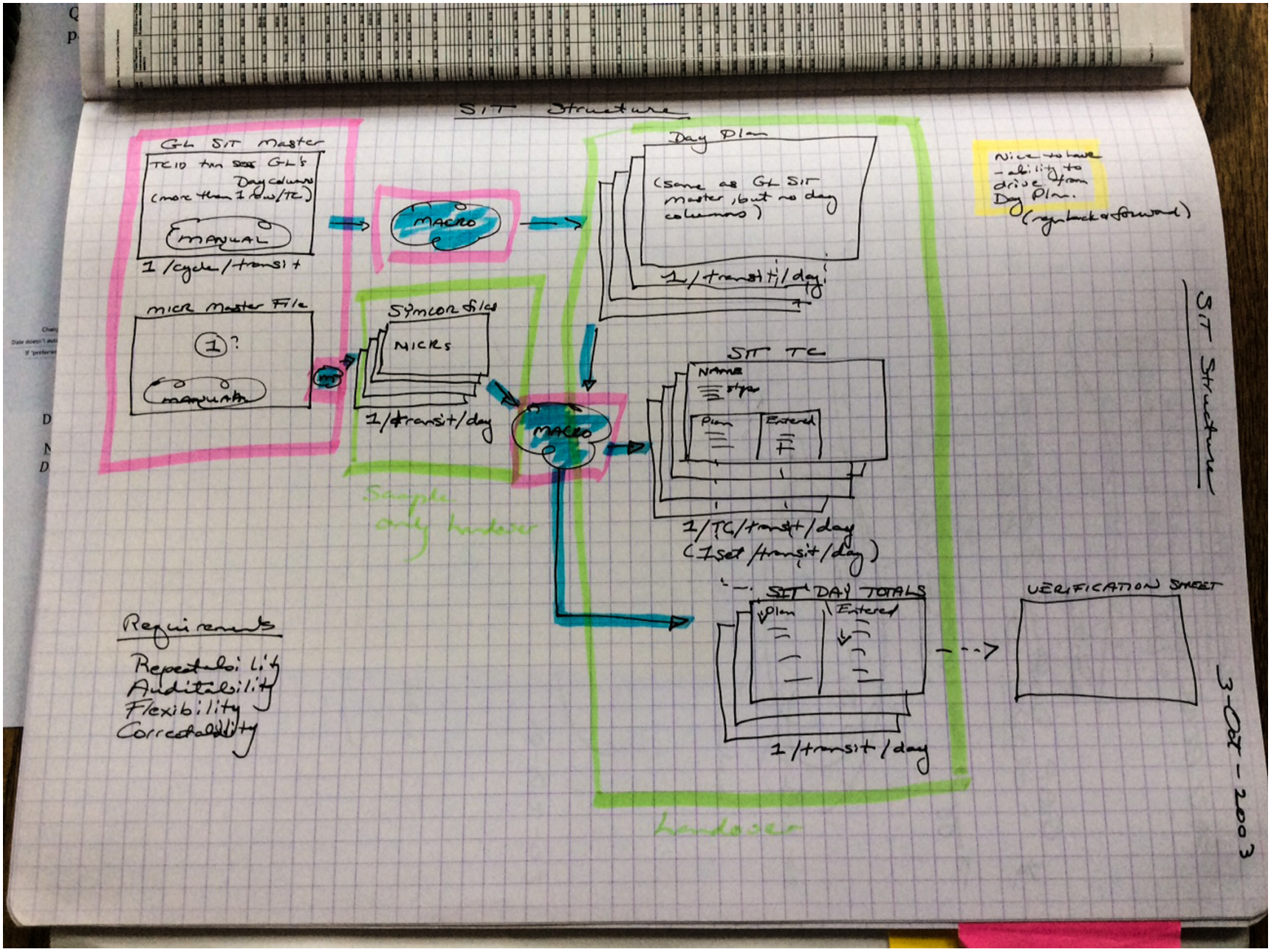


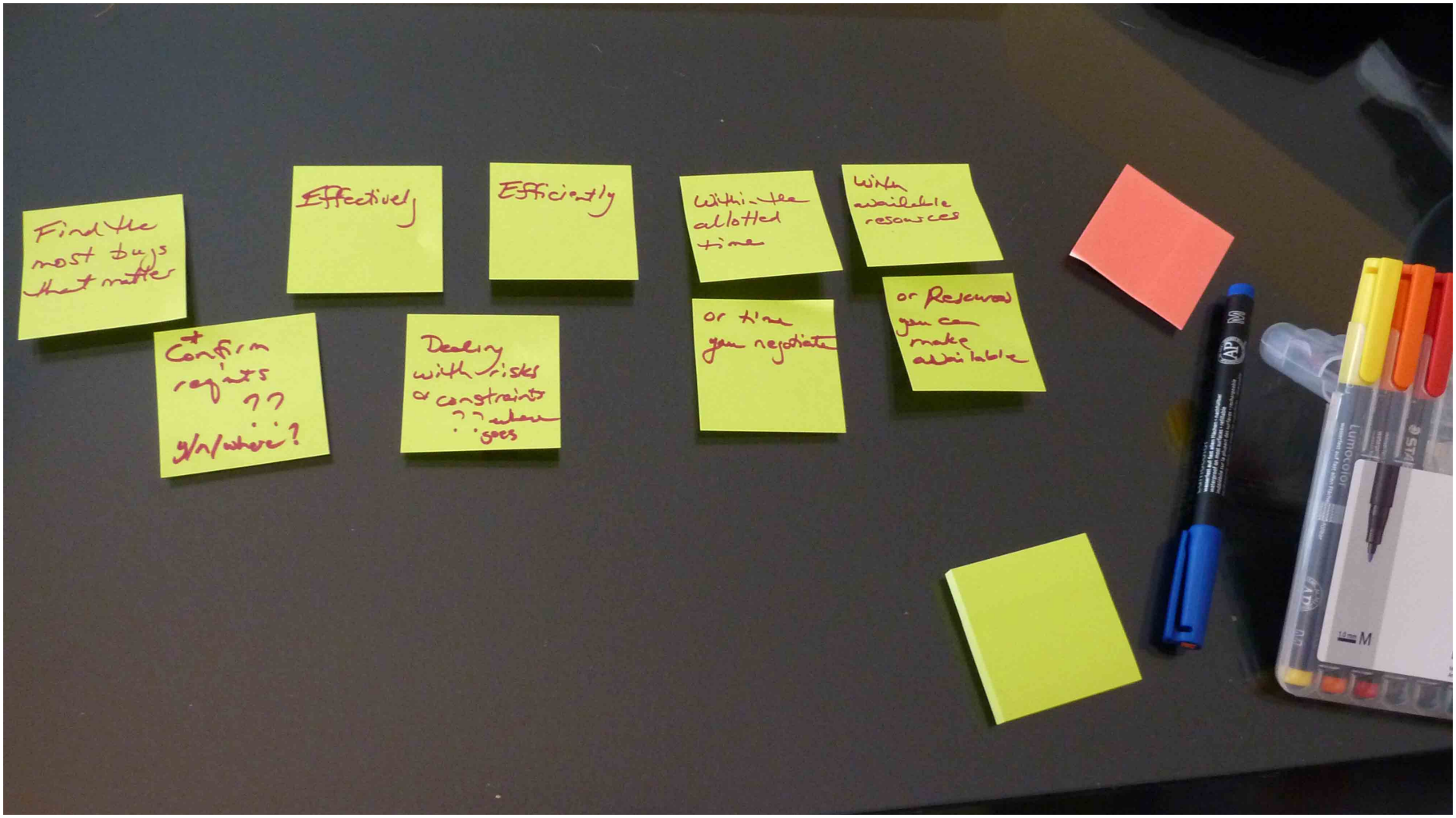
Teller Project
E2E SIT Test Artifacts

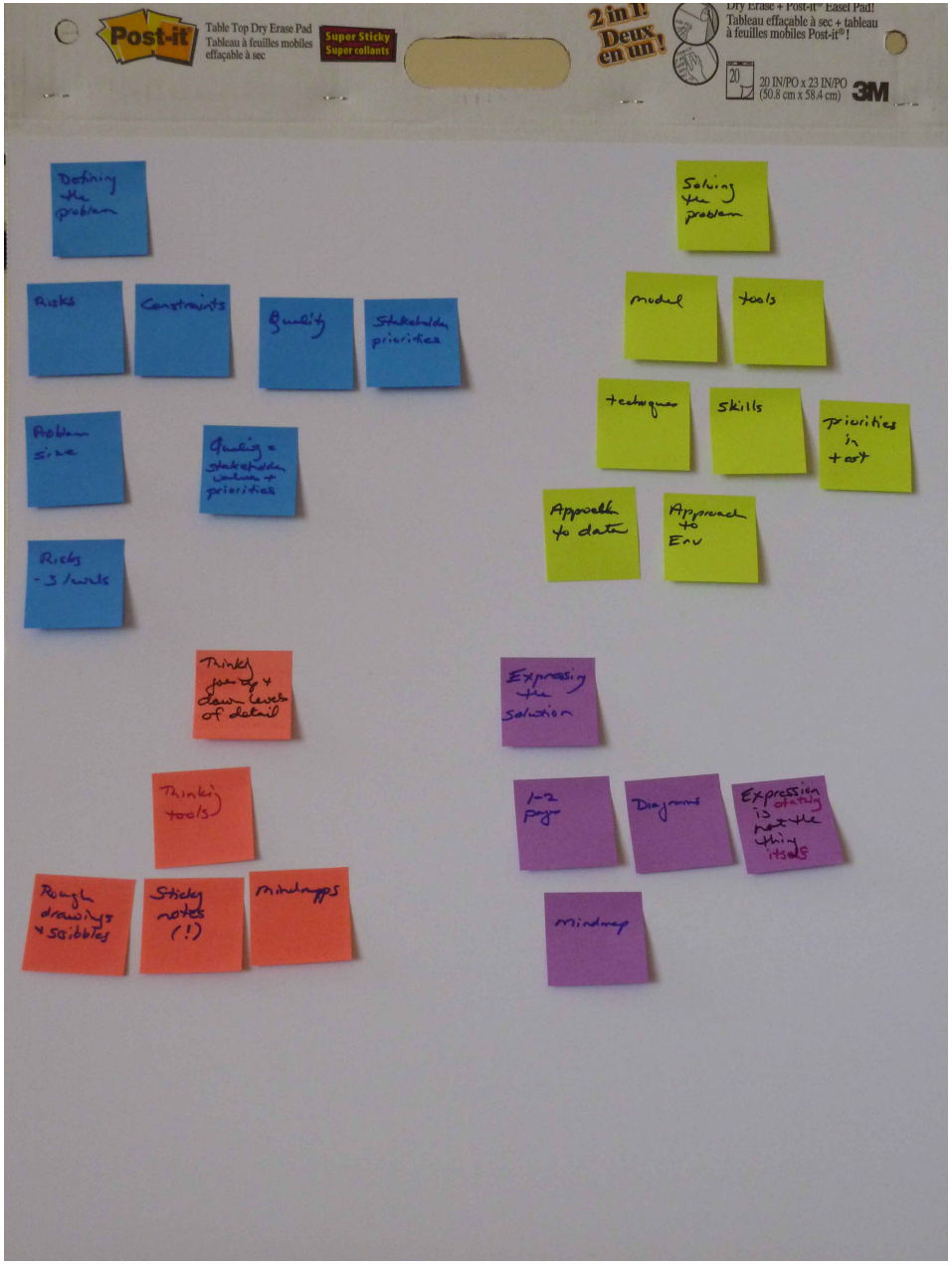


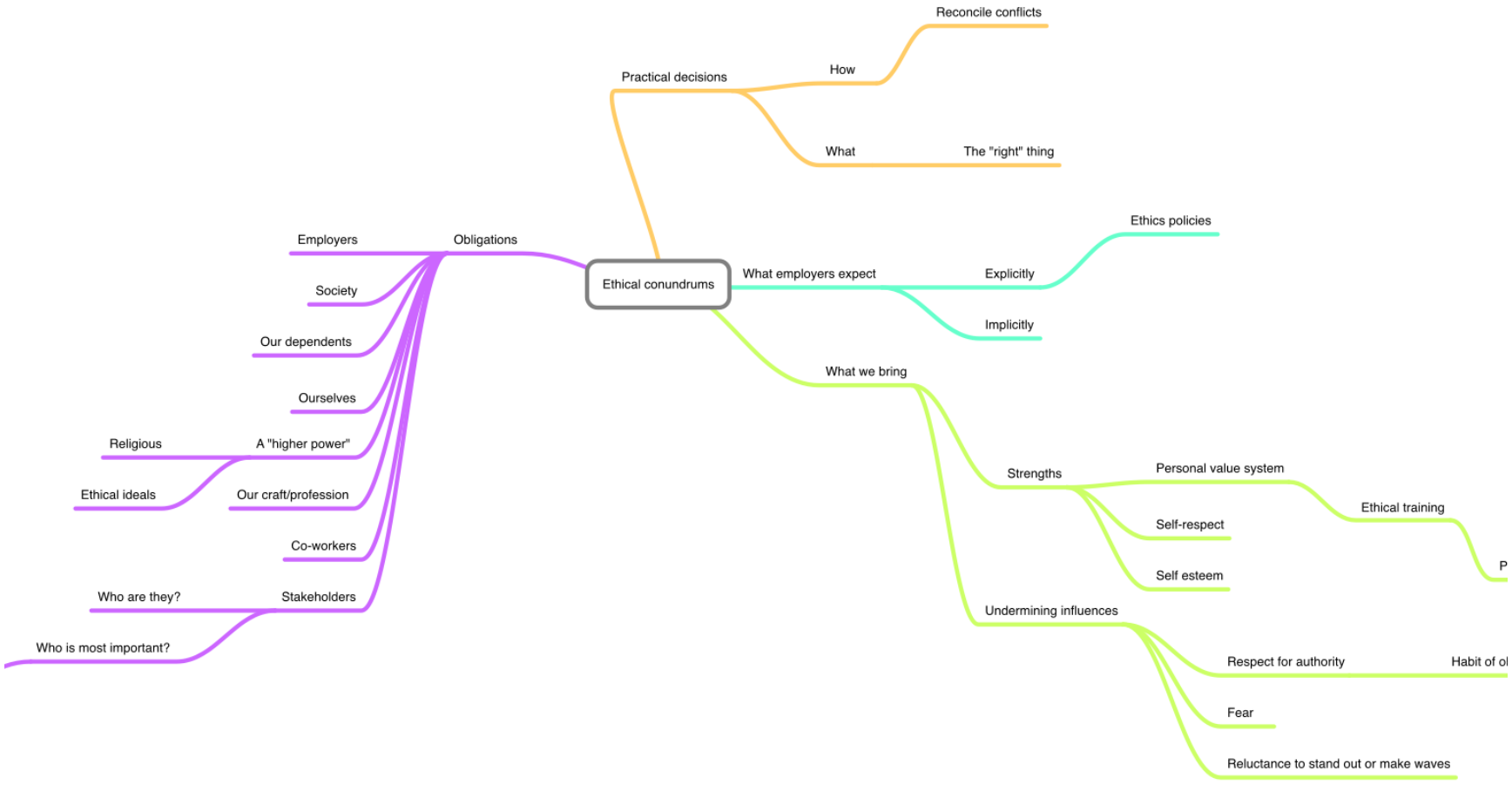
Getting there

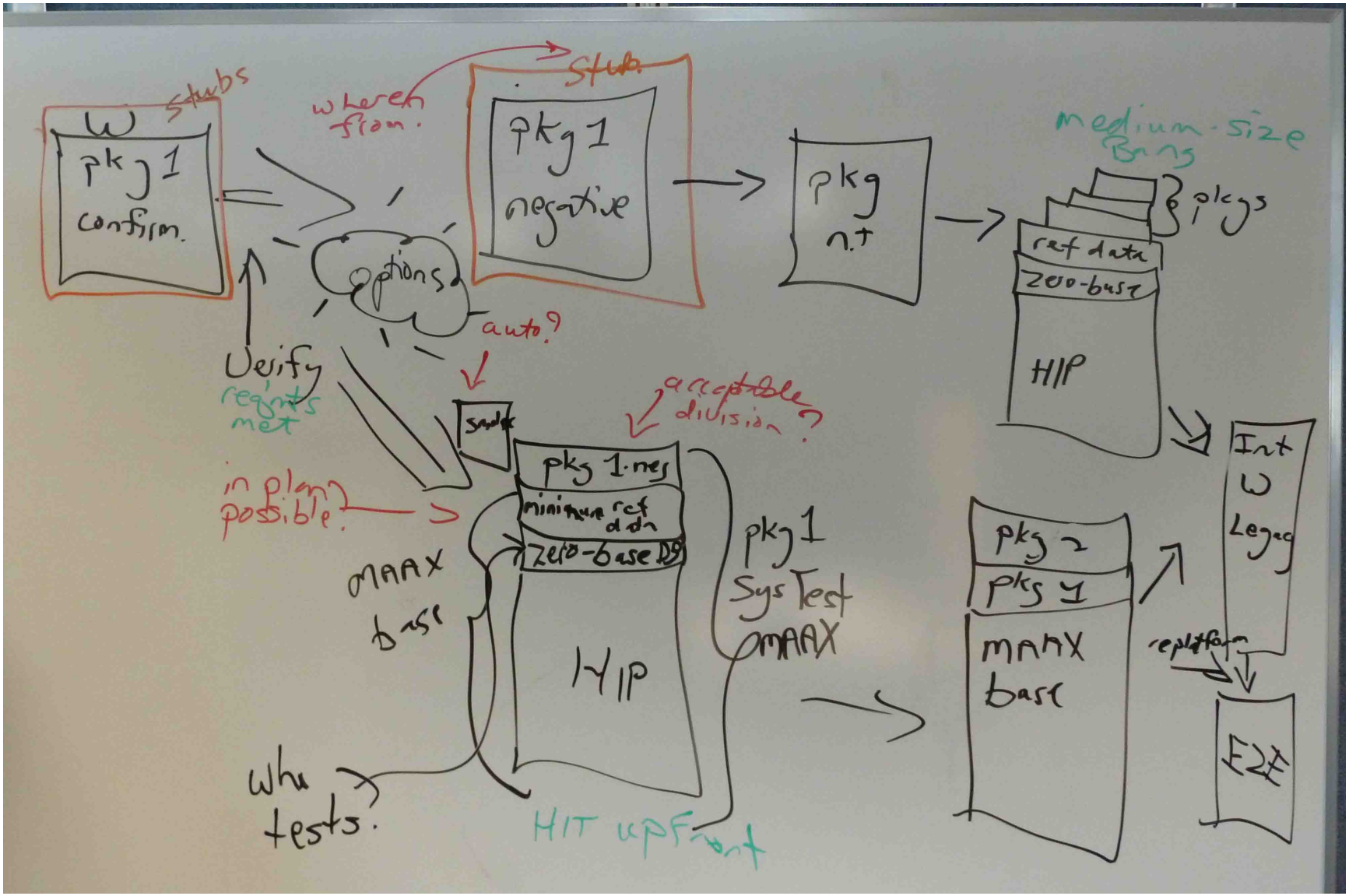




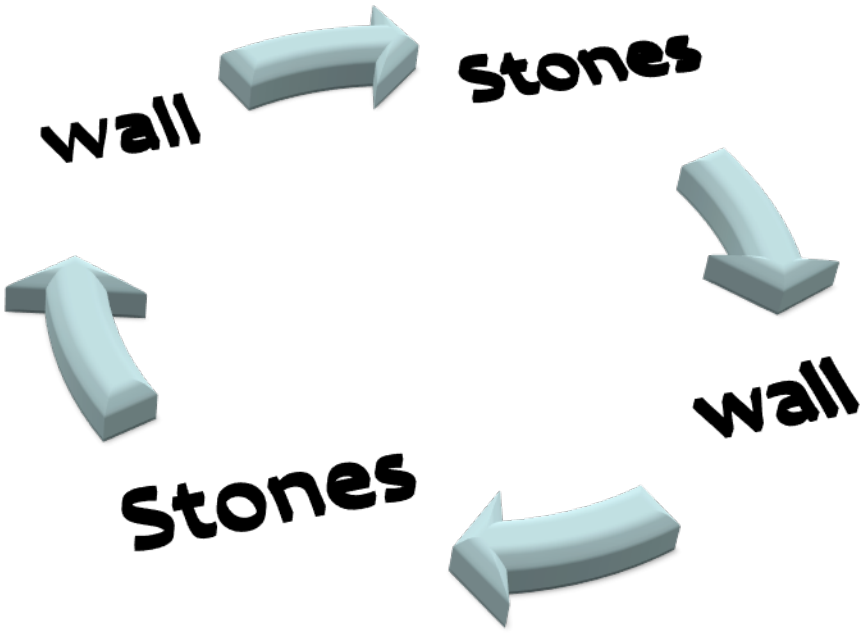








Work through levels of detail



Recap:

Understand the models you already use

- Strengths
- Limitations

Open up the range of models you could use

Use visual representations to understand and communicate your models and strategies

"… essentially, all models are wrong, but some are useful"

George Box

Fiona Charles

fiona.charles@quality-intelligence.com

www.quality-intelligence.com

Twitter: @FionaCCharles

Questions & discussion