Developing reliable and scalable decision support systems

Nordic MATLAB Expo
April 2016

Ariel Fischer
Head of Systems Development and Infrastructure
Trient Asset Management AS

Trient Asset Management AS

Independent asset manager

Fundamentally driven

Extensive use of quantitative analysis

Developing proprietary software

About me

10+ years experience in software development

First half «hacking», second half building systems

Felt the pain of writing «bad» code

Learned (hopefully) something along the way

Passionate about software engineering

Backdrop

38% of scientists spend at least one fifth of their time developing software

47% (only) of scientists have a good understanding of software testing

34% (only) of scientists see formal training in software development as important

Source:
«Why scientific programming does not compute»
Nature, October 2010

«There are terrifying statistics showing that almost all of what scientists know about coding is self-taught. They just don't know how bad they are.»

Greg Wilson
Nature, October 2010

Content

«It is not enough for code to work.»

Robert C. Martin
Co-author of the Agile Manifesto

Design Stamina Hypothesis

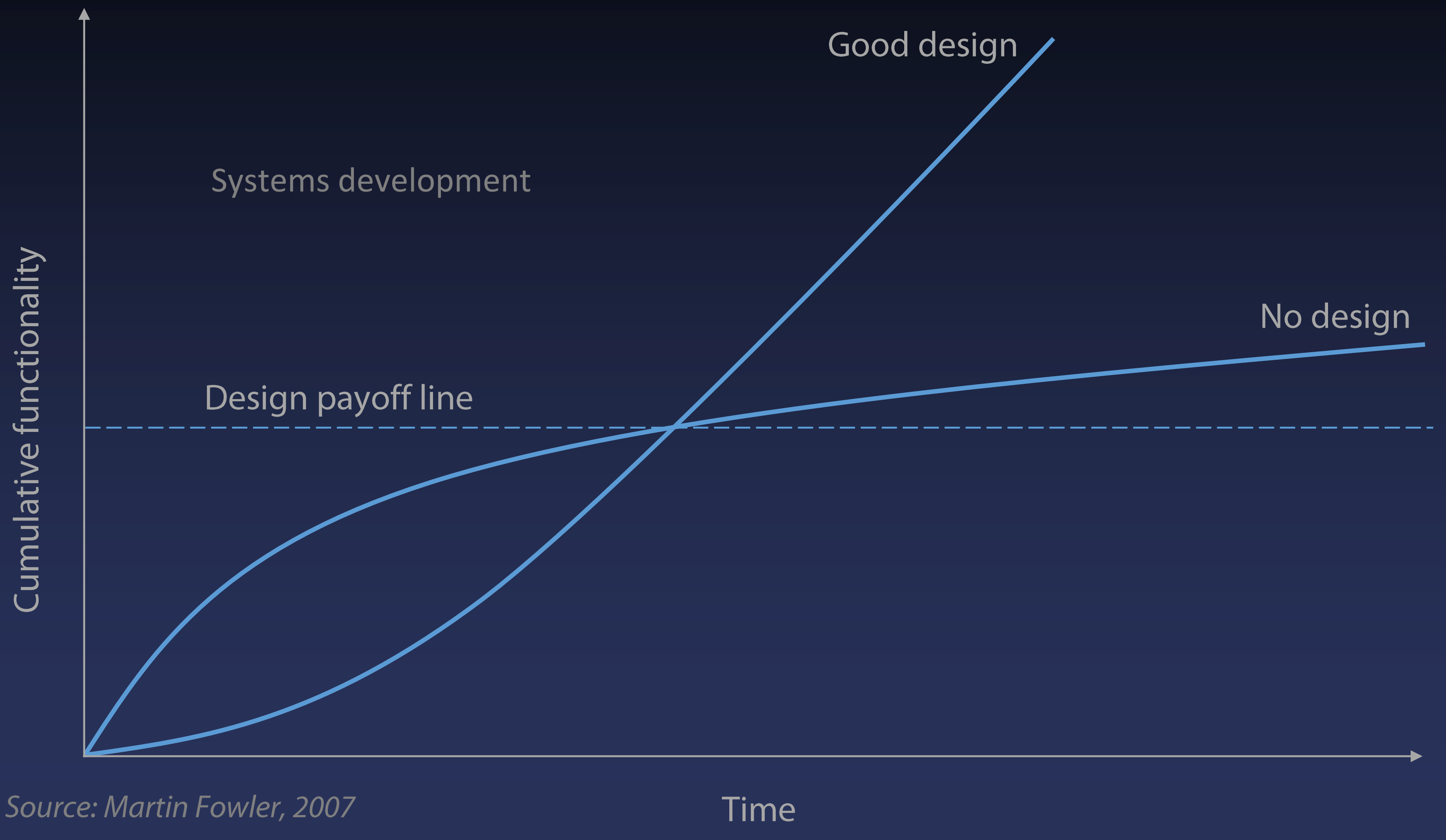
Important aspects of software development

Example of no design vs. good design

Focus: Maintainability and testability

Use case from Trient

Design Stamina Hypothesis



Source: Martin Fowler, 2007

Important aspects

of software development

Maintainability

Testability

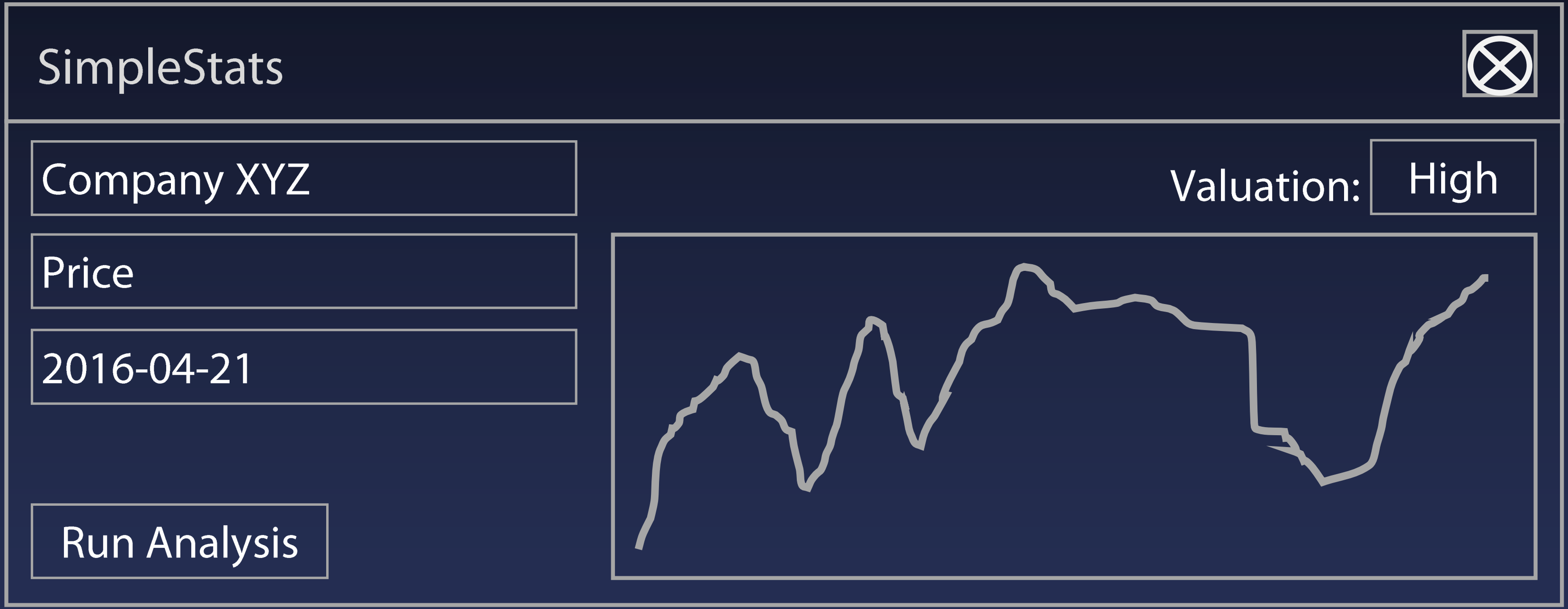
Reusability

Extensibility

Scalability

Reliability

Example



Company XYZ

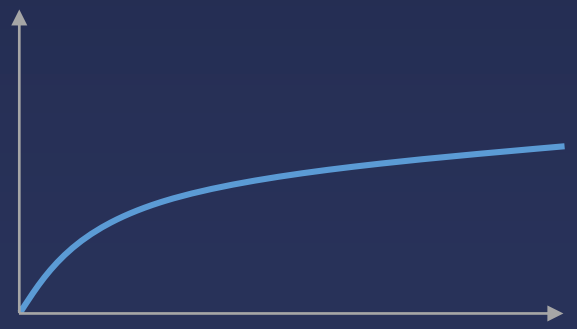
Price

2016-04-21

Valuation: High

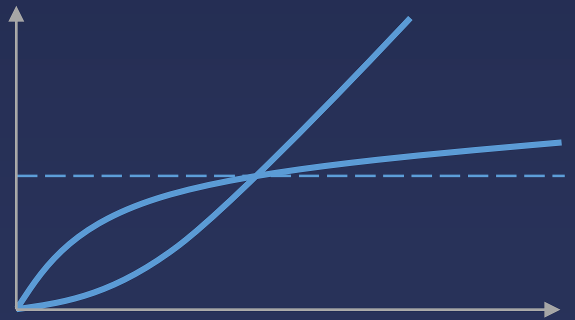
No design

```
function pushbuttonRunAnalysis_Callback(hObject, eventdata, handles)
  stockTicker = get(handles.textBoxStockTicker, 'String');
  field = get(handles.textBoxField, 'String');
  date = get(handles.textBoxDate, 'String');

  connection = database('CompanySQL', 'dbo', 'Password465', 'Vendor',...
    'Microsoft SQL Server', 'Server', 'MYSRVR4561', 'AuthType', 'Server','Portnumber', 1433);

  sqlStatement = sprintf(['SELECT Price, Date FROM TimeSeriesView WHERE ',...
    'Ticker = ''%s'' AND Field = ''%s'' AND Date <= ''%s'' ORDER BY Date ASC'],...
        stockTicker, field, date);

  cursor = exec(connection, sqlStatement); fetch(cursor); close(cursor);
  prices = cursor.Data.Price; dates = cursor.Data.Date;

  average = mean(prices); stdev = std(prices); score = (prices(end)-average)/stdev;
  valuation = 'In Range';
  if (score > 2)
    valuation = 'High';
  end
  if (score < 2)
    valuation = 'Low';
  end

  plot(datenum(dates), prices);
  set(handles.textBoxValuation, 'String', valuation);
end
```



With design

```
function pushbuttonRunAnalysis_Callback(hObject, eventdata, handles)
  stockTicker = get(handles.textBoxStockTicker, 'String');
  field = get(handles.textBoxField, 'String');
  date = get(handles.textBoxDate, 'String');

  prices = handles.DataStore.GetData(stockTicker, field, date);

  valuation = handles.SimpleStatsModel.GetValuation(prices.Price);

  plot(datenum(data.Dates), data.Prices);
  set(handles.textValuation, 'String', valuation);
end
```



Maintainability

The maintainability of a system (m) is inversely proportional to the coupling – the number of dependencies (n_d)

$$m \propto \frac{1}{n_d}$$

Testability

“Code without tests is bad code. It doesn't matter how well written it is; it doesn't matter how pretty or object-oriented or well encapsulated it is. With tests, we can change the behavior of our code quickly and verifiably. Without them, we really don't know if our code is getting better or worse.”

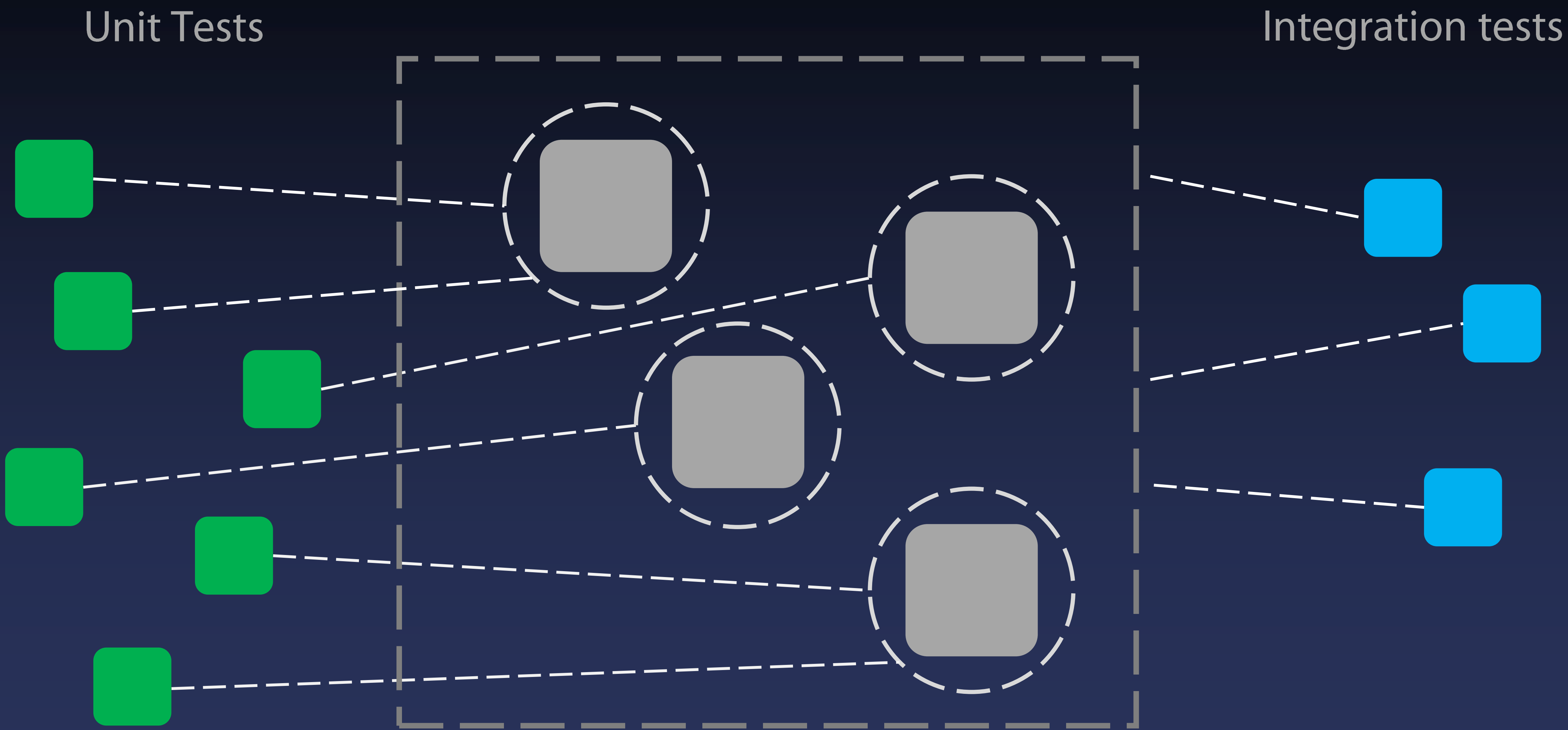
Michael Feathers
Author of Working Effectively with Legacy Code

Automated testing

Unit Tests
A small automated test, coded by a programmer, that verifies whether or not a small piece of production code – a unit – works as expected in isolation.

Integration Tests
An automated test in which individual software modules are combined and tested as a group.

Production Code



As of R2013a MATLAB has a built in framework for writing automated tests

Use case

Use case

Asset class models

Performance and attribution reports

Key takeaways

The design payoff line

Maintenance is inversely proportional to the number of dependencies

Code without tests is bad code

Thank you!

Ariel Fischer

Head of Systems Development
Trient Asset Management AS

(+ 47) 22 39 88 91
ariel.fischer@trientam.no
http://www.trientam.com

Founder and Entrepreneur
Turning Data Into Products AS

(+ 47) 995 98 547
ariel.fischer@quantifio.no
http://www.quantifio.no