Agile Requirements

Introducing User Stories

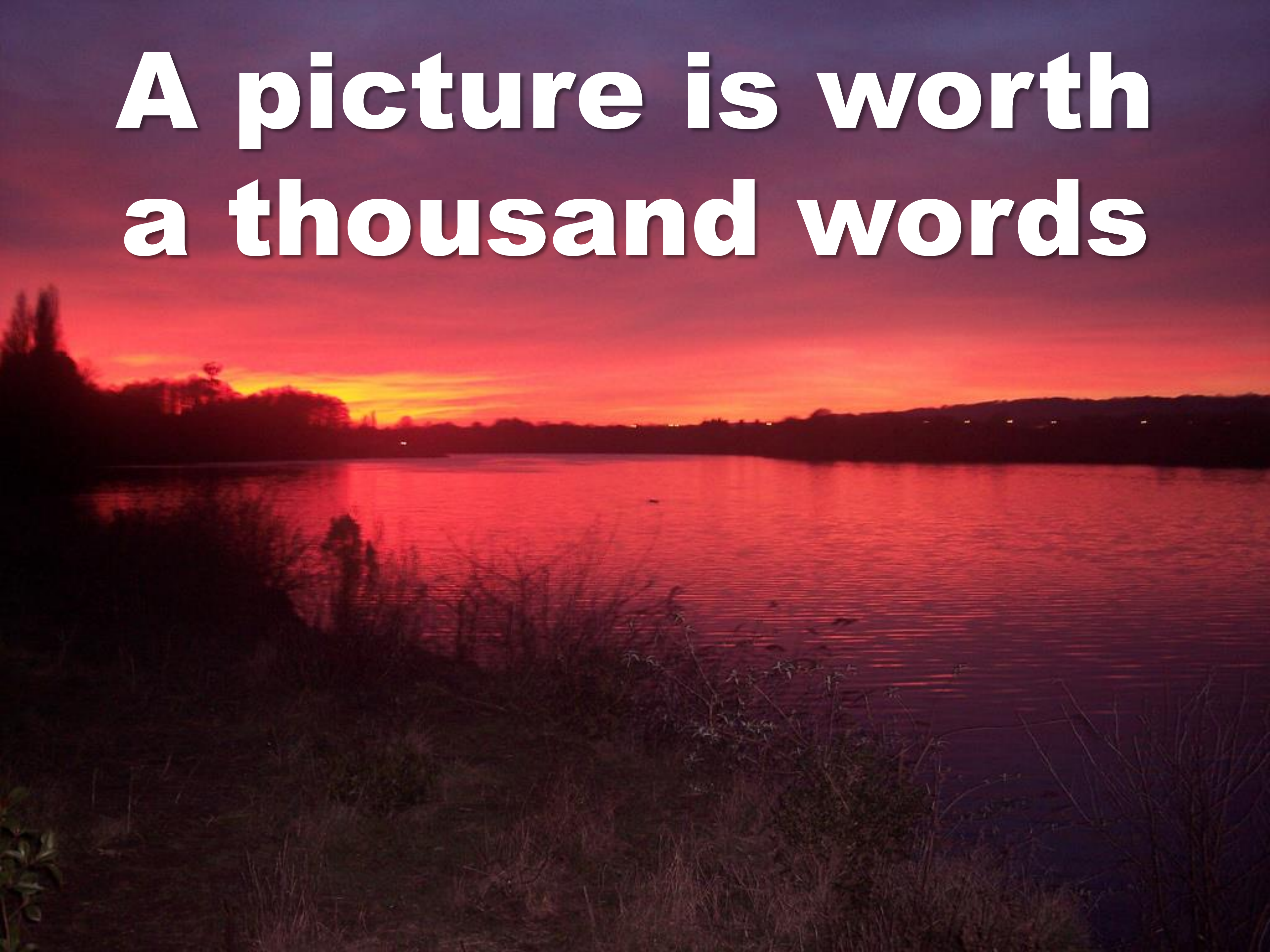
Key Principles for Agile Requirements

- Active user involvement is imperative
- Agile teams must be empowered to make decisions
- Requirements emerge and evolve as software is developed
- Agile requirements are ‘barely sufficient’
- Requirements are developed in small, bite-sized pieces
- Enough’s enough – apply the 80/20 rule
- Cooperation, collaboration and communication between all team members is essential

Requirements are a Communication Problem

- **Written requirements**
  - can be well thought through, reviewed and edited
  - provide a permanent record
  - are more easily shared with groups of people
  - time consuming to produce
  - may be less relevant or superseded over time
  - can be easily misinterpreted
- **Verbal requirements**
  - instantaneous feedback and clarification
  - information-packed exchange
  - easier to clarify and gain common understanding
  - more easily adapted to any new information known at the time
  - can spark ideas about problems and opportunities

A picture is worth a thousand words



User Stories

seek to combine the strengths
of written and verbal communication,
where possible supported by a picture.

What is a User Story?

A concise, written description
of a piece of functionality
that will be valuable to a user
(or owner) of the software.

User Story Cards have 3 parts

1. **Card** - A written description of the user story for planning purposes and as a reminder

2. **Conversation** - A section for capturing further information about the user story and details of any conversations

3. **Confirmation** - A section to convey what tests will be carried out to confirm the user story is complete and working as expected

User Story Description

As a [user role] I want to [goal] so I can [reason]

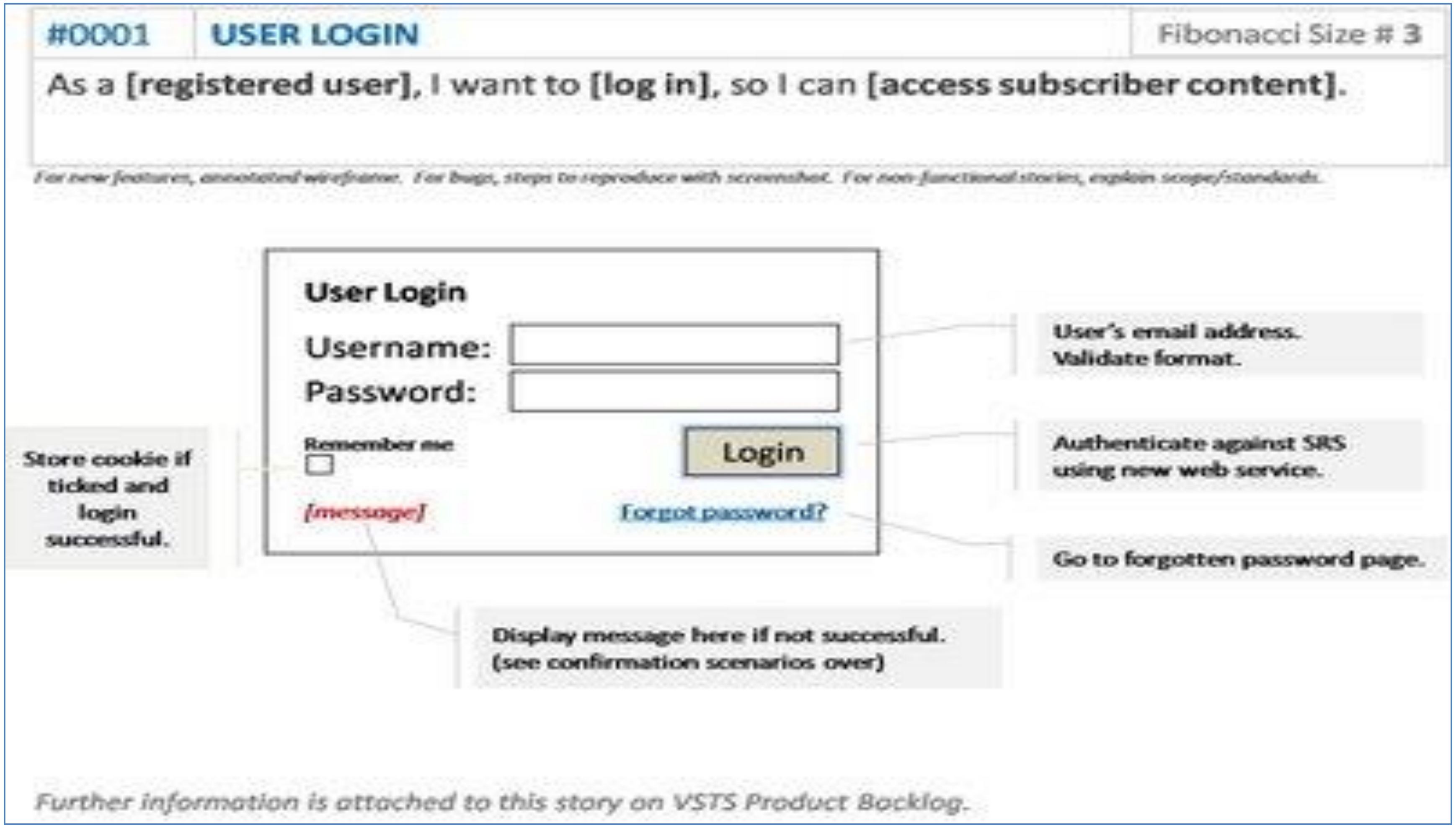
For example:

- As a registered user I want to log in so I can access subscriber-only content

User Story Description

- **Who** (user role)
- **What** (goal)
- **Why** (reason)
  - gives clarity as to why a feature is useful
  - can influence how a feature should function
  - can give you ideas for other useful features that support the user's goals

User Story Example: Front of Card



#0001 **USER LOGIN** Fibonacci Size # 3

As a **[registered user]**, I want to **[log in]**, so I can **[access subscriber content]**.

For new features, annotated wireframe. For bugs, steps to reproduce with screenshot. For non-functional stories, explain scope/standards.

User Login

Username:

Password:

Remember me

Login

[message]

Forgot password?

User's email address. Validate format.

Authenticate against SRS using new web service.

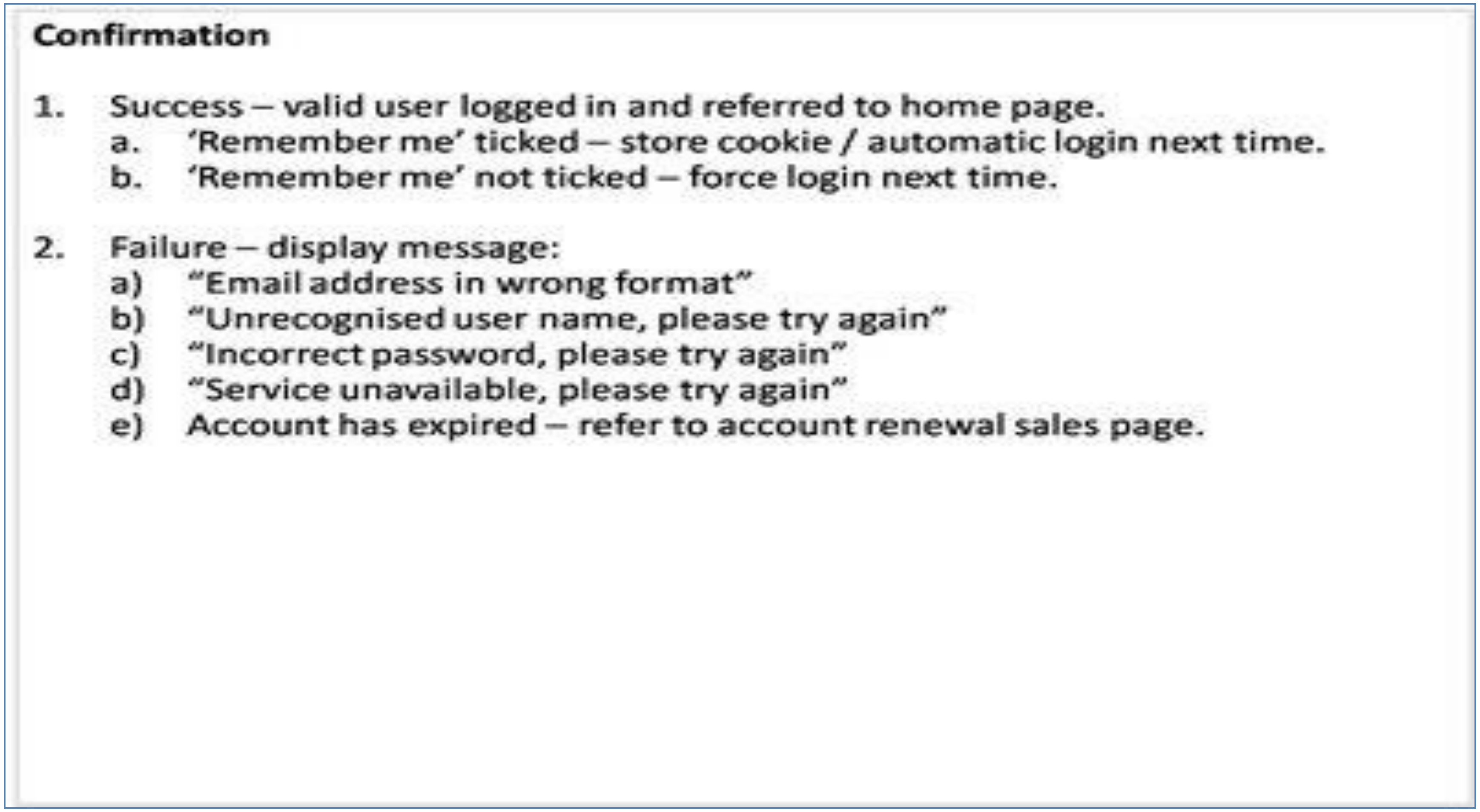
Go to forgotten password page.

Store cookie if ticked and login successful.

Display message here if not successful. (see confirmation scenarios over)

Further information is attached to this story on VSTS Product Backlog.

User Story Example: Back of Card



Confirmation

1. Success – valid user logged in and referred to home page.
   a. 'Remember me' ticked – store cookie / automatic login next time.
   b. 'Remember me' not ticked – force login next time.

2. Failure – display message:
   a) "Email address in wrong format"
   b) "Unrecognised user name, please try again"
   c) "Incorrect password, please try again"
   d) "Service unavailable, please try again"
   e) Account has expired – refer to account renewal sales page.

How detailed should a User Story be?

Detailed enough for the team to start work from, and further details to be established and clarified at the time of development.

INVEST in Good User Stories

- **I**ndependent – User Stories should be as independent as possible.
- **N**egotiable – User Stories are not a contract. They are not detailed specifications. They are reminders of features for the team to discuss and collaborate to clarify the details near the time of development.
- **V**aluable – User Stories should be valuable to the user (or owner) of the solution. They should be written in user language. They should be features, not tasks.
- **E**stimatable – User Stories need to be possible to estimate. They need to provide enough information to estimate, without being too detailed.
- **S**mall – User Stories should be small. Not too small. But not too big.
- **T**estable – User Stories need to be worded in a way that is testable, i.e. not too subjective and to provide clear details of how the User Story will be tested.

User Stories Summary

- User Stories combine written and verbal communications, supported with a picture where possible.
- User Stories should describe features that are of value to the user, written in a user’s language.
- User Stories detail just enough information and no more.
- Details are deferred and captured through collaboration just in time for development.
- Test cases should be written before development, when the User Story is written.
- User Stories should be Independent, Negotiable, Valuable, Estimatable, Small and Testable.

Writing Good User Stories

http://www.agile-software-development.com/2008/04/writing-good-user-stories.html