Agile/Traditional differences:

Traditional treats scope as king. Anything outside the scope is to be ignored, team actively tries to maintain project with the same scope.

Agile sees change as good, looks to develop the scope as the project proceeds.

Agile uses the MoSCoW principle (get Moscow table)

(Diagram comparing trad/agile (triangle w/scope,time,cost)

MoSCoW (must have, should have, could have, wont have (right now))

8 principles of agile (non-negotiable):

* Focus on the Business Need
  + Decisions based around project goals, not wants
  + Clear understanding of business priorities (Moscow rules)
  + Guarantee Minimum Useable SubseT (MUST haves)
  + Supported by business roles and agreed products
* Deliver on Time
  + Team must Timebox the work
  + Deliver according to MoSCoW
  + Never move agreed timeline
* Collaborate
  + Involve the right stakeholder at the right time throughout the project
  + Team members are empowered to make decisions
  + Actively involve business representatives
  + Hold facilitated workshops
* Never Compromise Quality (Fit for purpose)
  + Team must set and agree acceptance criteria
  + Ensure that quality does not become a variable
  + Design document and test
  + Test early and often
  + Constant review by the right people
* Build Incrementally from firm foundations (BIFO)
  + Team must strive for early delivery of business benefit where possible
  + Continually confirm that the correct solution is being built
  + Formally re-assess MoSCoW and ongoing project viability at each timebox
* Develop iteratively
  + Nothing built perfectly the first time
  + Enough Design Up Front (EDUF) just enough design to be able to make something that can be shown
  + Embrace and expect change
  + Agile knows that change is inevitable – harness the benefits
* Communicate continuously and clearly
  + Must run daily stand up meetings
  + Must use facilitated workshops
  + Must model and prototype
  + Must keep documentation
  + Must have face to face communication at all levels of stakeholder management
* Demonstrate control
  + Must guarantee timeboxing
  + Must use burn down charts
  + Must track velocity
  + Must use technical expertise

Time boxing: (check diagram, lecture 10)

Timebox is a planned portion of work

Each timebox has a set time (max 14 days)

Team does what they need to do in the timebox

Never compromise on quality

Never move to the next timebox until the previous timebox is complete

Plan next timebox at end of each timebox, including what to do and time

Identify, plan, evolve, review

Using Moscow

Must haves are required, should have are great if there is time.

Project stops if problem with time box

Within each timebox:

Kick off Moscow  
 decide obejctives of timebox  
 determine if can be done in allocated time  
 Is it the right next step?  
 agree on acceptance criteria  
 analyse risks for this timebox

Investigation, refinement, consolidation  
 (investigation: details product to be delivered and quantities)  
 (refinement: bulk of development and testing)  
 (consolidation: tying up loose ends and ensuring products meet acceptance criteria

Close out  
 Formal acceptance of the timebox deliverables

Daily Stand ups

Daily

Facilitated by team leader/project manager

Each member says what they’ve been doing since the last standup, what they’re going to do before the next standup, any problems,risks, issues they’re having

Typically ~15 minutes

Moscow and daily stand ups

Risk is constantly evaluated

Risk is assessed in each timebox

Stakeholders can identify risk in an ongoing manner

Agile Estimations

Feasibility

Develop outline plan, decide basic needs < 10 requirements

(go/no go)

Foundations

Refined estimates based on deeper knowledge

Firm and enduring foundations are established (unchanging things throughout project)

Prioritise requirements list

Identify physical infrastructure (EDUF)

Describe likely development lifecycle

< 100 requirements

BADs, SADs, DADs

Business Area Definition

Should be created for any project where the proposed solution impacts on the way “business as usual” is conducted within an organisation

System Architecture Definition

Should be created for any project where there is a systems aspect to the solution.  
defines technical framework within which the solution will be developed and provides a high-level description of the architecture for that solution.

Development Approach Definition

Defines how the solution development team will develop and how they will assess the fitness for purpose of the solution

(all practices to be developed iteratively)

(go/no go)

Exploration

Iteratively and incrementally investigate detailed business requirements (translate into viable solution)

Create functional solution that demonstrably meets the needs of the business.

Only move to exploration once everything in foundations is completed

Engineering

Refine evolving solution from the exploration phase

(refine the products required to successfully operate and support the solution)

Make sure evolving solution meets original version

Make sure technical elements are sound

All stakeholders agree

Constantly work towards a deployable solution

Deployment

Agile gets products deployed quickly so as to get results

Deployed product can still be refined to improve their results

Post Product

Describe how the benefits have accrued (what has improved)

Describe the changes that have been made

Explain what has enabled formal acceptance to have been achieved