References

Rising, Linda. “Agile Methods: What’s it All About”
http://www.ddci.com/news

eXtreme Programming

Simplicity - use it throughout

Communication - talk to customers and colleagues

Feedback - test software and process continuously

Courage - classical methodologies use fear of it not working, XP uses the courage to trust it will work

XP Practices

On-site customer. The Customer must be on-site and available full-time.
Planning game. Customers set priorities and developers estimate effort. The result is a schedule that the developers buy in to and provides value to the customer.
Small releases. Provide early benefit to the customer and early feedback to the developer.
Acceptance tests. Customers write the tests that will show them that the system works.
Simple design. Travel light. Don't produce anything extra. Do enough to satisfy your current needs but no more. Don't anticipate problems or future demands on the system.
Pair programming. One member of the pair is the implementer, thinking about here and now, while the other is working on strategy.
Unit test. Write unit tests before code. The all unit tests must run at 100% to show that nothing has been broken. If any test fails, you must stop and fix the problem.
Refactoring. After the code is written and unit tests pass, then make the code simpler. Use small, low-risk steps. Don't refactor on speculation.
Collective ownership. Anyone can improve any piece of code. Everyone is responsible for everything.
Coding standards. Express individuality in the way you wear your XP ball cap, not in your code.
Continuous integration. Integration testing happens at least daily when developers add their new code to the base. This forces a natural rhythm to development: test -> code -> release. All unit tests and all functional tests must pass.
Metaphor. The team should have a shared vision of the system. This ensures conceptual integrity.
Sustainable pace. No more than one consecutive week of overtime.

XP Resources

http://www.xprogramming.com/
http://www.objectmentor.com/publications/articlesByDate.html
http://www.extremeprogramming.org/

Scrum

project backlog:a list of all deliverables. These can be user stories (as in XP) or requirements. The backlog is never empty as long as the program is maintained. It constantly changes. It is updated by only one person, the designated project owner, who also sets the priority of the items. The backlog is always visible and anyone can influence the project owner.

sprint: 2-4 week division of a project. All stake holders meet to determine what will be done in the next Sprint. Create the project backlog.

During a Sprint, no interruptions are allowed from the outside. A short meeting is held daily or every other day where the team lead or **ScrumMaster**, who asks three questions of every team member:

- **What have you completed, relative to the team backlog, since the last Scrum meeting?**
- **What held you up?**
- **What do you plan to do between now and the next Scrum meeting**?

End of Sprint onto Next

At the end of the Sprint the stakeholders meeting is held.

- **Increments are delivered.**
- **Surprises are reported.**
- **ANYTHING can be changed.**
- **New estimates and team assignments are made for the next Sprint.**
- **The project can be cancelled.**

Scrum Resources

http://www.controlchaos.com/
http://members.cox.net/risingl1/articles/index.html

Crystal

human-powered and adaptive - achieving project success, people centric

ultralight - reduce paper to a minimum for the parameters of the project

shrink-to-fit software - start with something small

non-jealous - feel free to borrow from other methodologies

Crystal Process

Set up the first increment, learn, set up the next increment, and so on. Give the process to the team and allow them to change it. Will not tell you exactly what to do except:

- Collaborate closely
- Observe the process closely
- Iterate

Crystal Resources

http://crystalmethodologies.org/index.htm
http://members.aol.com/humansandt/crystal/clear/intro.html

Adaptive Software Development

- Speculate
- Conduct the project initiation phase
- Determine the project timebox
- Determine the optimal number of cycles and the timebox for each
- Write an objective statement for each cycle
- Assign primary components to each cycle
- Assign technology and support components to each cycle
- Develop a project task list
- Collaborate. Apply concurrent component engineering
- Learn. Use a quality review.
- From the customer's perspective
- From a technical perspective
- Functioning of the delivery team and their practices
- Project status

Adaptive Software Resources

http://www.adaptivesd.com/index.html
http://www.adaptivesd.com/publicat.htm