Validated - users are happy with the software product

Verified - Software is done right i.e. matches a specific set of requirements

Managed - Above requirements are done right (managed right)

What does the software manager need to know

- Manage people (the coders, the customers)
- Know the customer business case
- Know the software like a real user

Agile - Guidelines on how to develop software so best results occur

Scrum - An agile methodology

Defining success - often used schedule, budget and meeting user requirements. Also pay attention to developer satisfaction and so many other things

Studies have shown the the **most important thing is timing of the release**-see TED talk by Bill Gross.

Mandatory reading
Please read the two online articles mentioned below.

Scott Ambler writes in his blog article "There is No Common Definition of Software Development Success", that projects differ in how they define success; this is supported by his survey "2013 IT Project Success Rates Survey Results".

The survey also shows how modern methodologies such as Lean (e.g., Kanban), Agile (e.g., Scrum), and Iterative (e.g., Unified) are apparently more effective than Traditional (e.g., Waterfall) or Ad Hoc approaches.

These methodologies are further described in the Software Processes and Agile Practices course in the Software Product Management specialization.

http://scottambler.com/no-common-definition-of-success.html

http://www.ambysoft.com/surveys/success2013.html#Downloads

Agile manifesto (www.agilemanifesto.org)

4 very broad values

Individuals and interaction > processes and tools

Working software > Comprehensive documentation

Customer collaboration > Contract negotiation

Responding to change > Following a plan

12 more specific guidelines that follow from the above
grouped into 3 categories

A. Delivering working software - keep developing working demos and improve on them all the time.

1. Our highest priority is to satisfy customer through early and continuous delivery of valuable software. (valuable is defined by customer)

2. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

3. Working software is the primary measure of success. (it should be “complete” in Agile this means that all the code has been tested and documented. Better to complete a few features thoroughly rather than lots of incomplete features.

B. Flexible design and adapting to change.

4. Welcome changing requirements even late in development. Agile processes harness change for the customer’s competitive advantage.

5. Continuous attention to technical excellence and good design enhances agility. (write readable, simple code with flexible design)

6. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely - keep a pace of work that will be sustainable in the long-term. Do not burn people out.

7. Simplicity, the art of minimizing the amount of work not done, is essential. Write as little code and documentation as is possible while still getting the work done.

C. Collaborative Communication and Organization.

8. Build projects around motivated individuals, give them the environment and support they need and trust them to get the job done. Step back and let them do your job.

9. The best architectures, requirements and designs emerge from self-organizing teams. Assigning tasks, finding the right tools. (no leaders as such in the teams, all are equal)

10. Business people must work together daily, throughout the project.

11. The most efficient and effective method for conveying information to and within a development team is face-to-face conversation.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Processes
Phases -specification, design and implementation, verification and validation

Requirements
Why do you need them - avoids confusion.

Planning (e.g. Alan Leikin's work)
Plan what needs to be done and when, who does what and how long it will take, plan tasks and schedules, time estimates. Better requirements lead to better time estimates.
Plan for potential risks, problems and action plans should they occur.

Monitoring the project
Monitor, analyze and review your progress. Velocity is a good measure of progress, transparency means that everyone knows what is going on and not just "management" (aka VIsibility).