Twenty Ways to Split Stories

The Big Picture		
Research	Implement	*What have others done?*
Spike	Implement	*Explore a quick solution*
Manual	Automated	*Often have to retain manual solution anyway*
Buy	Build	*Can go either way; trade cost of customizing*
Build	Buy	*..versus cost of implementing yourself*
Single-User	Multi-User	*Fewer worries about scale, user accounts*
API only	User Interface	*Tests may function without user interfaces*
Character or Script UI	GUI	*Simple interface can prove out ideas*
Generic UI	Custom UI	*"Naked Objects" approach can be cheaper*
Ilities		
Static	Dynamic	*Do once and ignore updates*
Ignore Errors	Handle Errors	*Minimize error code (don't ignore exceptions)*
Transient	Persistent	*Focus on behavior over persistence*
Low Fidelity	High Fidelity	*Quality of result (e.g., pixel depth)*
Unreliable	Reliable	*"Perfect uptime is very expensive."–Wm. Pietri*
Small Scale	Large Scale	*Build load capacity over time*
Less "ilities"	More "ilities"	*Address non-functional requirements later*
Features		
Few Features	Many Features	*Easier to do fewer features*
Main Flow	Alternative Flows	*Happy path vs. all possible paths*
0	1	*Nothing is easier than something*
1	Many	*One is easier than a bunch*
One Level	All Levels	*One level is the base case for all levels*
Base Case	General Case	*Base case must be done; others needn't*

See full article, "Twenty Ways to Split Stories," http://xp123.com/xplor/xp0512