



How to Make Sure Your Implementation Fails (in 10 easy steps)

Greg Weichhand

August 17, 2012

Failure is simply the opportunity to begin again, this time more intelligently.

[Henry Ford](#)



1. Do not define failure

- If you do not define failure, no one can prove you did not fail.
- If you must define failure, make it as subjective as possible.
- If your organization wants to use KPIs, identify them late in the process.



2. Plan on the “best-case” scenario

- There’s always the possibility nothing will go wrong.
- Easier approval of budget and schedule.
- Once you have commitment, it’s easier to get more budget and schedule approved.



3. Avoid leadership commitment

- Inclusion of leadership slows down the decision making process.
- If sponsorship is required, keep it superficial.
- Provide the leadership with deniability when your implementation fails.



4. Do not involve business partners in decision making

- It's an IT project so they should be IT decisions.
- Business partners will thank you for not taking up their time with selection and set-up decisions.
- End users will have to use whatever system you give them anyway.



5. Keep the scope as flexible as possible

- Changes are inevitable, plan for them.
- Once end users test the system they will want more changes.



6. Keep communication to a minimum

- Best not to inform people until you absolutely have to.
- Minimizes disruption.
- Until you are ready to test, business partners and end users have other things to work on.



7. Customize, customize, customize

- Your organization is unique.
- You already know what are best practices.
- Changing processes is a lengthy, arduous and unwelcome venture.



8. Keep your implementation team one-dimensional

- Business team members know the business.
- Systems experts know the system.
- Having both leads to disagreements and competing interests.



9. Don't waste resources on “soft skills”.

- Efforts like Training and Change Management add time and cost to an implementation.
- People are used to change – the strong will adapt.
- Good end users will figure out how to use the system – it's their job.



10. Go live no matter what

- There will always be excuses to delay.
- It is easier to debug and fix a working system once you are live than to delay implementation based on unproven concerns.



My great concern is not whether you have failed,
but whether you are content with your failure.

Abraham Lincoln

