



Printable Version:

Use the print command from the menu above to print this item.

[Close](#)

STICKYMINDS ORIGINAL



Developing a Requirements Team

By David Gelperin

Summary: Forming, Storming, Norming, and Performing. No, they are not a law firm, but the stages of team development proposed by Bruce Tuckman in 1965. I wish I had known about this model much earlier in my career. If you haven't encountered it before and want to help your requirements team, read on.

Part of the challenge of requirements work is transforming a diverse group of stakeholders into a productive and efficient team. Tuckman proposed a staged model of team development that has stood the test of time. Understanding this model can improve the quality of stakeholder communication and lessen the time spent developing an effective team.

In brief, the team development stages are:

1. Forming

the team is meeting and greeting
everyone is polite
some stakeholders are excited
there is no obvious conflict
nothing important is decided

2. Storming

differences surface in viewpoints, priorities, and style
confusion and disagreements about team objectives, priorities, organization, and strategies are obvious
suspicions and impatience appear about team members and the project misunderstandings are frequent
there is little progress

3. Norming

common understanding and respect begins
criticism is expressed thoughtfully and constructively
consensus forms on work rules, goals, boundaries, roles, and strategy
improved listening and open sharing
a foundation for cooperation is created

4. Performing

cooperative relationships are established
the focus is on major issues and important tasks
team pride increases
work is getting done
substantial progress is made

Unless a team has successfully worked together or is homogeneous in outlook, style, and understanding, there is no way to skip the Storming and Norming stages. Requirements efforts rarely satisfy these skipping criteria.

This model certainly describes my experience with teams of diverse stakeholders. For example, from personal mistakes, I now understand that it matters a great deal when proposals are introduced to the team. The team may reject a proposal, for example about work product development strategies, introduced in the Forming stage before context and trust have been established. Rejection often includes wasteful discussions filled with misunderstandings.

If the same proposal is introduced in the Norming stage, the team may accept it with little discussion. Effective communication requires that the receiver be ready for the message. Now I know that warm bodies in a room is not a sufficient indicator of readiness to receive.

The relative calm of the forming stage provides the team leader with an opportunity to define expectations and behavioral ground rules that can help the team negotiate the remaining stages. An expectation is that the team will learn a great deal during requirements development. The team will never know less than it does at the beginning. Admitting that requirements development is a journey out of ignorance by means of an exercise in group learning helps the team deal with the doubts of development.

You may find it effective to charge each stakeholder with two major responsibilities: to learn and to teach. Stakeholders must not only learn about requirements, but about each others viewpoints, priorities, and styles. This learning needs active listening and suppression of quick judgments.

The teaching responsibility entails sharing experience and expertise to enable other team members to understand some of the reasons behind suggestions made by the member who is teaching. Without minimal teaching, suggestions become commandments and the expert must always be right. Without teaching, there is little room for discussion on many issues and when needed, negotiated compromise is difficult since no side can understand the reasoning of the other. Team acceptance of their learning and teaching responsibilities smoothes the transition through Storming and Norming.

A variant of the Tuckman model is Forming, Storming, Ignoring, and Pretending. During these stages:

1. **Forming** – [Same as Tuchman model]
2. **Storming** – Uncovers or should uncover major conflicts in goals, priorities, or strategy among critical stakeholders
3. **Ignoring** – Fails to either resolve the conflicts or halt the project
4. **Pretending** – Follows the protocols of requirements development with little hope of project success

Do these models describe any of your requirements experiences? Let us hear from you.

About the Author David Gelperin is CTO & President of LiveSpecs Software in Minneapolis, MN. He has more than thirty-five years experience in software engineering with an emphasis on software quality, verification, and testing (SQVT) and software process engineering. David has been a SQVT consultant/mentor and instructor (20 yrs), quality support manager (5 yrs), verification lead (2 yrs), project lead (2 yrs), and programmer (5 yrs). He has consulted for both commercial and in-house software development organizations. In 1986, David co-founded Software Quality Engineering (www.sqe.com, www.stickyminds.com), the leading provider of software quality information worldwide. In addition, David chaired the development of both ANSI/IEEE standards on software testing – 829 on software test documentation and 1008 on software unit testing.

[Close](#)

Software Quality Engineering Copyright 2004 ©