

Discussion Brainstorming Techniques

A Possible Chautauqua Handout for my Proposed Session

Steve Chenoweth
Rose-Hulman Institute of Technology

Here are guidelines and explanations for some of the techniques especially useful in brainstorming activities related to software engineering (SE).

The 3-D model

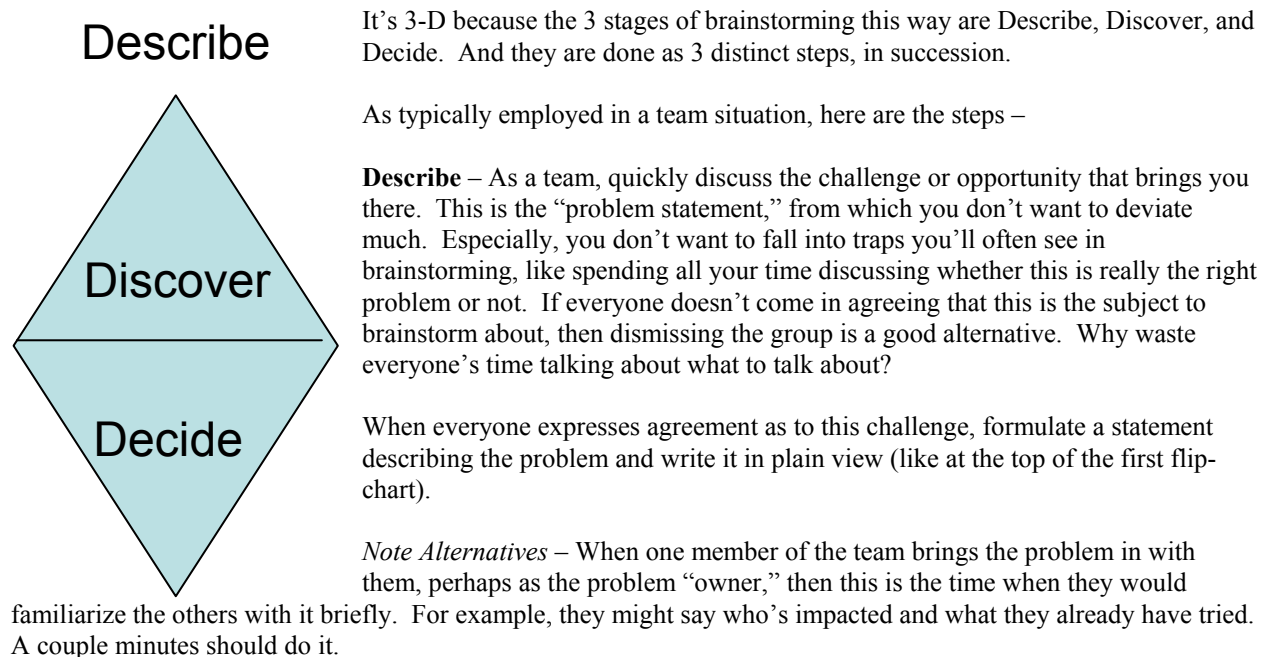
What it is: The 3-D model is an example of the “diamond” style of creative problem-solving, discussed in many references. It was developed by Jordan Ayan of Create-It, Inc., Naperville, IL. For lots of additional brainstorming ideas, see Jordan’s book, *Aha! 10 Ways to Free Your Creative Spirit and Find Your Great Ideas*, available from Amazon for \$ 10.20.

Something it’s good for in software engineering: Making sure you’ve looked seriously at design alternatives before committing to one. (Plus lots of other situations – it’s a style of general purpose problem solving.) Why good for exploring design choices? Because engineers tend to commit, in their thinking, to focusing just on a particular solution way before they have to, leaving other options on the table.

Prereq knowledge: For design, you should at least understand the short version of the problem to be solved.

Expected outcomes: Possibly multiple alternatives, or at least the chance to review them once again before picking the leading one. Also, documentation of all those other choices, in case you need to revisit this territory.

How to do it: Here’s the 3-D model for brainstorming:



Another useful trick is to see if someone can finish a phrase like “We’ll be done when ...” Jordan Ayan recommends putting the challenge in the following manner:

Imagine [whatever the problem is, only now solved, looking back on it]. What ideas will have contributed to the success?

It also can be helpful for someone to characterize the desired solution qualitatively. By default, some of these values come from the organization and people at work. For example, which of these is it?

“We want a creative, new solution.”

“We want only a low risk solution.”

“We want a long-term solution, even if that takes extra work now.”

“We want a fast, short-term solution.”

“We want *any* solution, but this is a really hard problem.”

With a group which already works well together in brainstorming, you *can* get by with fuzzier problems. If they are new, better make the problem really clear. Brainstorming itself also tends to fuzz the problem somewhat. Say, you’re looking for a new way to keep the pot holes filled in your town, and one brainstormer suggests, “I wish we could fly – then we wouldn’t care about pot holes.” That’s fine, because it opens people up to thinking in other directions. However, if someone goes, “I think this whole pot hole question is a waste of time,” that’s going to close everyone up, and get them off track. So, keeping people on task during brainstorming is an art form.

In one of our the brainstorming methods suggested *below* (Chindogu), the problem itself may unclear. Thus, we did one 3-D process starting with the fuzzy problem and ending with a clear problem to work on. Then we did another 3-D process to solve it!

If you want other helpful ideas about expressing problem statements, check with me.

Discover – This is the part that we emphasize in brainstorming, coming up with lots of new ideas. It is at the heart of creative engineering and of synthesizing new products and services. Discussed *below* the 3-D model are some of the various methods we may try in class, or you might want to try yourself.

In brainstorming, a worthy general goal is for a team to press on *beyond* where it becomes difficult to come-up with more new ideas. This is where the aha! ideas usually occur, with extra, unplanned work.

Beyond this, all the usual ideas apply, regarding what makes for good idea generation during brainstorming:

1. No premature evaluation
2. Wild ideas are welcome
3. Don’t wait
4. Quantity is key
5. Piggyback onto other ideas you hear.

Usually, it’s a good idea to have a facilitator for the whole activity, but especially for this brainstorming part of it – This would be someone whose sole job it is to run the brainstorming session, to keep order, and to record what everyone says. The facilitator has a key role in making sure everyone is involved (the mechanics of which vary depending on the method of brainstorming). Note also that traditionally the facilitator does not contribute their own ideas, even though this always is tempting for them. This means that, in a group like a software engineering development team with a manager, probably the manager should *not* lead the brainstorming, because he/she will naturally want to contribute ideas. Since it’s hard for anyone else who works for them to keep the manager in-line and participating as an equal during brainstorming, tradition calls for someone outside the manager’s group to be the facilitator.

Decide – Sort and select the best ideas, using one or more processes that help make this work well.

Sorting: In the middle of brainstorming, there are methods to organize what you've got, so it's not just a big long list of crazy ideas. We will likely use the *Affinity Diagram* in class, so you can see how it works. Mostly, you arrange post-it notes into groups. That's described in more depth, *below*.

Selecting: Don't forget that the emerging ideas need to fit whatever values you began with! For example, if one goal was to be innovative, and everyone now picks only easy targets to proceed with, then you've just managed to fool yourselves!

One way to decide democratically is to use *Dot Voting*. That is described separately, *below*. We used consensus to decide in one case – getting everyone to agree works a bit differently. Many times, the choices made after brainstorming also may be totally the opposite of consensus – like, the boss or client decides what they want to proceed with.

Conclusions and follow-up – Most people feel that a problem-solving session should end with one or more *emerging ideas* that people in the meeting agree really could be acted upon. It is very useful to end a session with a clear list of the ideas people expect to expend resources on, and a short, agreed-upon *action plan* for that. Getting this action plan might itself involve another 3-D process! I guess we would have added “Action Plan” at the bottom point of the diamond, except that it didn't start with a “D.”

It also is useful to end a problem-solving session with a bit of time to reflect on how the processes went, with ideas on what to try next time.

Special environmental concerns: In the top part of the diamond, we are doing divergent thinking in a group environment. Good divergent thinking requires that you loosen as many constraints which usually guide your thinking. For this reason, use of a facilitator, doing it off-campus, and requiring that people not do regular work (like taking cell phone calls) at the same time, all add to the quality of the result.

How to evaluate results: *Divergent part* -- By far the most important measure of any brainstorming exercise is how many ideas you got in a given period of time. Most people err horribly on the side of having too few, usually by waiting until some refinement is done before they start writing them down. Good facilitators don't make this mistake. *Convergent part* -- Usually the measure of value in *use* of the divergent knowledge is that the ideas converged upon remain interesting. There may or may not be a consensus – often, requiring a consensus flattens the result – the fact there are strong opinions may in fact indicate a higher level of interest.

Brainstorming generally

Group brainstorming has a classic form and some common variations, described in some of the sections *below*. Keep in mind that brainstorming exercises also have side effects, namely:

- Building teamwork spirit, through sharing of each other's less refined ideas in a structured way
- Learning to use tools that stimulate team ideas and help solve real creative problems
- Breaking out of limiting patterns when individuals can't think of solutions

Thus, the philosophy and techniques and motion and feeling we all put into practicing these methods were built around these goals. Keep in mind that “breaking out of the box” is an important tool to get innovative designs and processes. It also is a difficult task for any group of software engineers, given our usual penchant for polishing technical details.

Remember that whenever you are working on a project in the future and feel blocked, the 3-D process and these tools can be helpful. You also will gain from the shared motivation of working as peers on something creative, so you're way ahead if you've cultivated a team to brainstorm things with you.

The next bunch of topics are all “ways to brainstorm” and details of pieces of the 3-D diamond, above. There is no right way. Indeed, it may help a group be creative if they try brainstorming in some novel way, just to be different:

Technique – Idea Writing

What it is: This is “how to use the post-it notes” for brainstorming, in the “Discover” part of the 3-D model. Idea Writing is a form of idea generation allowing everyone to make a contribution, and generate a large volume of ideas in a short amount of time.

Something it’s good for in software engineering: Building ideas about the different parts of a system, or considerations about it (like what should be in hardware vs software, or what should be bought versus built). For example, what are all the tasks to consider in accomplishing the most difficult part of this new project? This typically requires lots of ideas from a mixed group of internal stakeholders – like systems engineers, architects, developers, testers, managers, and the product support group. Idea writing gets a huge number of ideas onto the table, to aid in spotting the one so tricky that everyone was trying not to think about it.

Prereq knowledge: A shared definition of the thing being explored.

Expected outcomes: A large number of ideas about the subject, which are in a form which is easy to organize as a next step.

How to do it: The process used for Idea Writing is quite simple. First, the group develops a statement of the problem or opportunity (as with the other brainstorming methods). Once the group agrees on this challenge or opportunity, they are ready to start generating ideas.

To do this, everyone is given a pad of 3” x 5”¹ post-it notes. Participants should work in silence and write each idea they have on a post-it note. One idea per sheet. Participants can generate as many ideas as possible in a given period of time, or until they run out of ideas.

Thoughts for the facilitator (note, *above*, that we recommend having one):

1. You can either have people write a whole pad-full of ideas, then everyone post them all in public view, or else have people post them as they think of them. Most groups prefer the first. You can even write ideas ahead of time, bringing them to the meeting. Those who lean toward one-at-a-time make this into a happening, where the facilitator runs around collecting and posting the ideas as participants tear them off their pads. Maybe the facilitator calls out the idea at that time to stimulate others’ thinking, maybe not. Definitely do *not* try to organize the ideas while brainstorming – that’s a different mode of thinking!
2. When used first, this technique tends to get out all the ideas everyone was thinking of already. A useful step is to challenge everyone at the end to “think of 2 more ideas.” Then “think of 1 more.” Here, even with this technique, one gets into the art of facilitation. The brainstorming works better if you can persist beyond the point where most people have, uncomfortably, run out of ideas. (Or think they’ve already solved the problem.)
3. Once people start to see what others have posted, they begin to think of more ideas. There’s no reason they can’t write those down! So, expect to get more ideas as you get to the next step, where people organize what you have.

Individual Idea Writing lends itself well to Affinity Diagrams, described next.

¹ That’s 7.5 cm x 12.5 cm. So, standard post-it notes of around this size.

A more daring *variant* of Idea Writing that sometimes works very well is Brain Writing, which uses **shared tablets** of paper for this silent brainstorming. This is described in a separate section, *below*.

Special environmental concerns: This method works better each time it’s used. If you try the variations, you’ll discover which one’s best for your group.

How to evaluate results: The number of post-it notes generated, and their variety. The latter may be a function of the variety in the group members themselves.

Expert application: The attached table summarizes some of the ways that idea writing contrasts with traditional brainstorming from a flip chart (described separately, later). Notice that each has advantages in certain situations!

Brainstorming method:	Advantages	Disadvantages	When to use	When not to use	Best practices
Traditional (with flip chart)	People listen to each other’s ideas.	They forget what they were going to say.	Group of 8 – 14 people having diverse viewpoints.	Too many divergent voices must be heard.	Ask for main idea first, write verbatim as fast as possible as each contributor expands on it. Move quickly to next.
	Builds group cohesiveness.	Participants need to be reminded of expected behavior.	A skilled facilitator is available.	Someone with existing strong group presence (like the boss) has to facilitate.	Write the problem statement on first page, to avoid divergence.
	Can see ideas easily on flip chart.	Room dynamics control contributions.	Participants used to sharing ideas and learning.	People afraid to talk openly, or unsure what to say.	Keep all flip chart pages visible in the room, so they can inspire more ideas.
Idea writing (with PostIt notes on a wall)	People can add their ideas very rapidly.	Less focused attention on others.	Groups of 4 – 14 contributors.	Contributors less likely to think of own new ideas.	Have contributors pre-write some ideas before meeting, on provided PostIts.
	Ideas can be offered more privately.	Some people have trouble writing concise ideas.	Organizing ideas is an important part of decision.	Building group cohesion is a key goal.	Announce each idea as its PostIt is added to the wall.
	Group can go back and organize PostIts as a next step.	Have to go up to wall to read ideas already posted.	Ideas can be explored further later on.	A decision or action plan is needed at the meeting.	Pick up PostIts in a well-defined order, write-up as an outline.

Table 1 – A comparison of traditional brainstorming and idea writing

Technique – Affinity Diagram

What it is: This is a technique for organizing ideas once you have done some brainstorming. It sits kind of *in the middle of the 3-D diamond*, in terms of when you’d use it. Works especially well with brainstorming methods like idea writing where you have each idea on a separate piece of paper. Let’s talk about it in relation to idea writing, as an example:

Something it’s good for in software engineering: Grouping the parts of a process or system into areas to explore next. You can take important details and combine them conceptually, which gives a bottom-up systems analysis.

Prereq knowledge: The people doing it should be the ones who just generated the ideas. So, they know some of what they're about to organize, and are invested in it.

Expected outcomes: An organized version of the results of brainstorming. This form is more amenable to converging on "what's important" out of those results.

How to do it: Once the idea writing process has been completed, an affinity diagram is a useful tool to group together similar ideas. Participants should congregate around the flip charts (or other designated posting area – white boards or even windows sometimes work well), and stick all of the post-it notes on the chart if this hasn't been done previously. As similar ideas are identified, they should be grouped together. When a common group is created, someone can post a *title* over the group that describes the ideas in the grouping. Participants should keep moving the post-it notes until the team arrives at a consensus that the groupings are correct.

Often, facilitators allow discussion during this activity. Sometimes when working on business problems it is useful to have the group create the affinity diagram without conversation.

Special environmental concerns: You need a surface on which post-its can be placed and replaced, without falling off, in an area where all the people working on it can read and handle the post-its.

How to evaluate results: The organization often is considered ideal if it finds a home for all the post-its, with the categories being of vaguely similar size. At the same time, most people usually put identical ideas on top of each other. So, the brainstorming outcome is now both structured and slightly smaller.

Technique – Dot Voting

What it is: This is part of "Decide," a technique that fits into *the bottom half of the 3-D diamond*, for choosing amongst all the ideas you've brainstormed. Note that it is not a consensus tool, but it *is* democratic!

Dot voting is a simple technique for helping a team or group arrive at a group decision when there are a variety of different options to choose from. When teams generate ideas, there are often hundreds of options. The dot voting technique helps a group narrow their options quickly and in a democratic manner.

Something it's good for in software engineering: Reaching decisions in a group of peers, or a group who want to act as peers for the initiation of a project. (E.g., the boss is in the exercise but, for this, gets the same number of votes as everyone else.) When shared vision or shared technical expertise should drive a decision, this works. In diverse groups, such as geographically distributed teams, dot voting can create a strong feeling of inclusion.

Prereq knowledge: Understanding what all the ideas mean, upon which one will vote.

Expected outcomes: Only the ideas getting the most votes survive for follow-up activities after the meeting.

How to do it: Each participant is given colored stick-on dots. How many? A good rule of thumb to use is to start by giving everyone three dots (or votes). Everyone can place their dots on the ideas that are recorded on the flip chart (or post-it notes) which they like the most. An individual should not vote for the same item more than once (i.e., if you have three dots, you must place them next to three different ideas). If this narrows it down to a handful of options, and you are trying to come up with one clear winner, have everyone vote a second time on the top vote-getters, this time using a different colored dot.

It is very useful before voting to restate the problem, and to restate as well the group's key underlying values. For example, "We are looking for a new, long-term solution, so we want one with some intrigue and difficulty, not just the easiest one."

Special environmental concerns: If the real action after brainstorming is that the boss is going to decide upon which items to explore further, then it makes more sense for them to do this right then, and not pretend to empower the group in this regard. This course of action is perfectly legitimate. It does not violate the spirit of brainstorming, so long as it's clear from the start of the process that this is how convergence will be done. Bosses should know that groups who feel they *did* participate in the actual decision making tend to work with more energy on making those decisions work. On geographically distributed teams, it also is important to understand that the democracy of dot voting is foreign to the way corporations operate in some parts of the world.

How to evaluate results: Most brainstorming professionals would say that the quality of the convergence is measured by the level of interest and energy in the room once the few have been selected from the many. A crucial side-event in using dot voting is that everyone feels they understand all the alternatives and acted together in picking the ideas to walk away with.

Technique – Consensus Building

What it is: This is an alternative to voting or mandating a choice, in the “Decide” part of the 3-D model.

Something it's good for in software engineering: Ensuring buy-in from all parties, critical when their follow-up cooperation is required for success of an overall project. On many software teams, the fact of reaching a consensus also is extremely motivating and energizing.

Prereq knowledge: Everyone must understand or appreciate the differing perspectives of the other people in the room who also will be voting.

Expected outcomes: The group speaks with a unified voice about how to proceed after the meeting, like a jury in a courtroom trying a criminal case.

How to do it: It is pretty hard to reach a consensus, even in a small group of brainstormers. Usually the team is chosen on purpose so as to have people with different points of view, so that you get many divergent ideas on how to solve your problem. So, how do you get them all to agree – which is what a consensus is?

The greatest success in this comes from working toward a consensus all along, not just when it comes time to make a decision. Examples of how to ease toward universal agreement can be found on the web, by searching for “consensus building.” See for example, <http://projects.edtech.sandi.net/staffdev/tpss99/processguides/consensus.html>.

A typical trick in consensus building is to stop and verify that everyone is in agreement all along the way, and spend time working out any differences.

Special environmental concerns: Everyone must believe that their views are valued, before they will “bend” to agree on something which they don't fully understand. So, for example, signs of valuing the participants must be present, which they'll pick up on as cues. Any overt acts by participants which show disrespect of others are likely to provoke “protest” acts when the group tries to achieve a consensus. As a result, such acts, or even personal disagreements which flare up, almost must be settled in front of the group before proceeding to convergence on ideas.

How to evaluate results: Reaching a consensus almost speaks for itself as an achievement!

Technique – Edsel

What it is: This is another brainstorming technique, for the top half of the 3-D diamond. It's distinctly different, because it does not ask participants to contribute just positive ideas! Its indirect path toward those results lets additional, less top-of-mind ideas emerge.

Something it's good for in software engineering: Describing the quality attributes for a second release of a product.

Prereq knowledge: Experience with the problems of the first release of that product.

Expected outcomes: A candid list of the overall traits required for a product to succeed.

How to do it: The Edsel technique takes its name from one of the greatest automotive marketing disasters. Ford produced the Edsel in the U.S. during the 1958-59 model years; both of these tries were clearly engineering, styling, sales and public relations failures. Ford discontinued the car, somewhat in disgrace, shortly after the 1960 model began delivery. Anyone who owned one of these cars will attest to the fact that it was just a bad idea.



Left: An ad image of the 1958 Edsel, from website www.fiftiesweb.com/pop/edsel.htm. Edsel collectors, of course, disagree with the rest of us about how great these cars were!

For the Edsel Technique, you will need someone on your team to act as the idea scribe at your team's flip chart. They should draw a vertical line down the center of the page.

The group should use traditional brainstorming to generate a list of the *worst* possible ideas that they can think of that address the issue they are focused on. Each of the ideas should be recorded on the left side of the flip chart. Usually, this first part of the process is humorous, because some of these bad ideas actually represent the first release of the product. After the team has run out of bad ideas, the team should go back over the ideas that were generated,

and try to *reverse* the bad ideas, turning them into good ideas. The good ideas should be recorded on the right side of the flip chart.

There's no need to get exactly one good idea for each bad idea, nor to do the good ones in the same order.

Special environmental concerns: The team must be free to critique their own product, as if it were the Edsel.

How to evaluate results: The list of quality attributes (or other results) should include surprises, things the team has not been expending effort to perfect, but should be.

Technique – Guided Visualization

What it is: This is a very powerful method often used to open people up *prior to brainstorming*, or even as a lead-in to just talking about a difficult subject. It's used in a wide range of places, like:

- sports psychology (picture yourself defeating Roger Federer in your next tennis match),
- business (closing a big deal with a key client),
- wellness and spirituality (imagine overcoming your chronic pain), and
- psychotherapy (picture yourself not afraid to get on the airplane).

Something it's good for in software engineering: In system design, very often this spellbinding activity is used to “picture what it's like being a customer,” or “picture the operators trying to use this system,” or picture some other role of people you need to interact with in your job.

Prereq knowledge: You actually don't want people already familiar with solutions to the problem they'll be visualizing. You want them to have to *try* to picture a solution – therein lies the power.

Expected outcomes: Someone in the room has a rush of ideas, and everyone has increased confidence in solving the problem.

How to do it: Plenty of guidelines and ideas for doing this are out on the web. And they do have varying recommendations and specifics about the techniques. See for example, the site <http://www.crystalinks.com/guidedvisualiza.html>, which talks about using guided visualization to connect ideas and events related to something of interest. I've tried to find the magic combination of words to Google for to get the best sources, and that's not easy. Guided visualization, imagery, and brainstorming, maybe.

Special environmental concerns: You need a group who are open enough with each other to engage in this activity. So, some warm-up building trust may be in order. Of course, we recommend someone facilitate the visualization who knows what they are doing. The problem is that this technique is used as a part of an awful lot of stuff, including different religions and self-improvement systems. Sorry about that. If you find additional good sites on this, please pass them along! And I don't recommend burning candles while you're doing it individually – just in case you fall asleep... The process also hypnotic – with luck, nobody thinks they're a chicken when they open their eyes.

How to evaluate results: The outcomes are going to be nebulous. But, that's the point – people become more receptive to new ideas via the visualization. It very much loosens people up for brainstorming, or for open discussions of issues.

Technique – Chindogu

What it is: This is another brainstorming technique, for the top half of the 3-D diamond:

This can be a kind of brainstorming game, too, because you are inventing something; but it's really a technique, a little like Edsel. In Chindogu, the brainstorming concept is to come-up with a solution that works, but nobody really would want it. By doing such things intentionally, we are almost automatically “out of the box” of thinking only about solutions that are fully acceptable by our current standards. Isn't it helpful to break out of there part of the time!

Something it's good for in software engineering: Deciding what you can pull out of Release 1.0 of a product, and still sell it.

Prereq knowledge: Strong familiarity with the subject.



Expected outcomes: A list of ways to achieve a plan, which are technically correct even if they may not be the most desirable.

How to do it: The dust-buster cat, *left*, pretty much sums up what Chindogus are. The rules of Chindogu and many examples can be found on the WWW. For instance, see web site where we found the cat, <http://website.lineone.net/~sobriety/>. Also, see the famous

books on the subject, *99 More Unuseless Japanese Inventions*, and its predecessor *101 Unuseless Japanese Inventions*, by Kenji Kawakami. W. W. Norton & Company, publishers.

As a technique in brainstorming for product development, Chindogu could be used like Edsel, above, but with the initial list to be Chindogu's. However, the goal of making a Chindogu can be much more elaborate, involving the full 3-D process. The intrigue and enjoyment of coming-up with something humorous and not-quite-right add to the value derived from playing this game. Oddly, Chindogu can work best when the outcome is very serious and must be done exactly right for success. Like, when you have to de-feature Release 1 of a software engineering product because you're running out of time – what can you take out of it and still have a marginally acceptable product?

Special environmental concerns: Having examples around from some of the above resources is very useful to inspire the Chindogu theme.

How to evaluate results: The resulting de-featured ideas about how to finish-up release 1.0 of a product may or may not be serious. The likely convergent stage to follow this divergent one would be to decide what shortcuts or changed results actually should be considered seriously by the team developing the product.

Technique – Traditional Brainstorming

This is another brainstorming technique, for the top half of the 3-D diamond. It's the one most people know as "brainstorming" generally. It has been attributed to [Alex Osborne](#), as a method of coming up with new ideas for advertising.

There actually are many variations to the basic brainstorming we've all used. And you might be surprised that not everyone does it the same way; for instance:

Free Flow – This is probably the most common technique where everyone shouts out an idea as they think of it. This can sometimes be difficult for the idea scribe if the pace is fast. It also opens the door for one person to monopolize the flow of ideas. If you are facilitating and you notice that one person is, for example, commenting on or adding to the value of almost everything being said by others, it may be time to switch to a different brainstorming technique!

Experienced facilitators usually modify free flow by calling on the less talkative team members to keep them involved, and by strictly limiting the elaboration of any one idea to what's necessary for identifying it on the flip-chart.

Round Robin – This technique is also sometimes called rotational brainstorming. Each participant contributes an idea in turn going around the table or room. Each person gets an opportunity to throw out an idea, but can also say "pass" if they so choose.

The success of Round Robin is quite sensitive to group size. One way to help keep everyone involved is to provide pencils and paper. When someone has an idea that reminds you of something, and it's not your turn next, you write it down so it's not forgotten. The success of getting people to do this writing as "filler" depends on many variables. Often, it works much better if there are several people in the room who are used to doing such writing, to serve as models. Everyone else sees them busy writing, and follows suit.

Eyes closed – If you want to increase the flow of ideas, have everyone close their eyes during the idea generation process. This will typically increase the idea output by about 30 percent because non-verbal cues (which create self-censorship) are eliminated.

Something it's good for in software engineering: Building group cohesion when a development team has just been put together, via building off each other's ideas.

Special environmental concerns: See Table 1, above, under Idea Writing, for how traditional brainstorming contrasts with that method as a major alternative. Note that “the boss” usually is an especially poor choice as the facilitator to write things down on the flip chart, because they tend to filter ideas as they write, as a part of their normal role.

How to evaluate results: In team building, the main question is, did everyone contribute to the significant outcomes of the activity?

Technique – Brain Writing

This one is easy to use in a large or small class. Basically, it involves passing pads of paper around, with people successively adding ideas to the ones already there. Here are the details:

Something it’s good for in software engineering: Analyzing the costs and impacts of bad quality – fishing for the big ones!

Prereq knowledge: Requirements you have; plus how the requirements imply that the proposed software system should work.

Expected outcomes: Surprise impacts which no one person would have anticipated. This method carries with it the silent social interaction of note-passing.

How to do it: Sit in circle (8-10 people), each person holding a pad of paper. To begin, each person thinks of a problem or other quality impact which they believe could occur with this design. Everyone quickly writes their idea at the top of the top sheet on their pad. Then the pads are passed to the left.

Each person silently reads the prior idea written by the last person, then thinks of a new problem or impact, related or unrelated, and writes that below it on the same sheet. The pads are passed to the left again, and the reading and writing process is repeated to create another new entry on each sheet.

When each pad has (simultaneously) gone around the whole circle, the person holding a particular pad silently reads the 8-10 ideas on the sheet they began. They circle the three most interesting ones contributed by others, using “surprise” as the criterion. They then put a star next to the one which sounds “surprising and most important.”

Each person takes turns reading the one surprising and important impact which they starred, explaining why they believe this is an important impact to study for the proposed software system.

Note: Don’t forget also to do a systematic analysis of issues! This method is designed as a check on that, to avoid those surprises.

Environmental concerns: It gives you results that aren’t as easily posted and sorted as in Idea Writing, described *above*. This can be overcome partly by having people identify ideas for further development, as noted here.

Technique – Biographical Brainstorming

This is another brainstorming technique, for the top half of the 3-D diamond:

What it is: Biographical Brainstorming is a forced association technique where you make connections by looking at your problem or opportunity from the perspective of a historical, current, or fictional character. That is, a team tries to look at their problem from the perspective of others. There are two variations that work well. One is to have everyone on the team brainstorm as if they are taking on the role of the character. The second method is to

have everyone brainstorm together from the perspective of the selected character. Write all ideas on the flip chart as they are expressed.

If you are not familiar with, or are not comfortable with a character, move on to another. Here are some sample characters you might use:

Agatha Christie	Frank Lloyd Wright	Muhammad Ali
Al Capone	Franklin Delano Roosevelt	Napoleon
Albert Einstein	Galileo	Norman Rockwell
Alicia Keys	General Norman Schwarzkopf	Oprah
Amelia Earhart	Hank Aaron	Outkast
Andy Warhol	Harriet Tubman	Ray Charles
Ann Landers	Helen Keller	Richard Simmons
Arnold Schwarzenegger	Henry Kissinger	Rosa Parks
Beethoven	Jacques Cousteau	Salvador Dali
Bill Gates	Jeff Gordon	Shakespeare
Carl Sagan	Juan Ponce de Leon	Sherlock Holmes
Cecil B. DeMille	Julius Caesar	Sitting Bull
Christopher Columbus	Kramer	Snow White
Cleopatra	Lucy and Ethel	Socrates
Colombo	Madonna	Steve Jobs
Daniel Boone	Mae West	Susan B. Anthony
Dick Clark	Martha Graham	Thurgood Marshall
Donald Trump	Martha Stewart	Tom Cruise
Dr. Dre	Martin Luther King	Walt Disney
Dr. Seuss	Mary Poppins	Will Smith
Eddie Murphy	Miss Universe	Winston Churchill
Evil Knievel	Mr. Potato Head	

For more ideas on people to use for such things, Goggle for Famous People, etc. I have a 1-page biography on each of several hundred people, some more famous than others; these can be used as “background” study, to “put these people to work on your project,” etc.

What it’s good for in software engineering: Getting developers “out of the box” after they’ve been working together for some time. Development teams notoriously get “group think” where they believe they are being creative but they are missing things that, say, competitors might think of.

Technique – Picture Connect

This is another brainstorming technique, for the top half of the 3-D diamond:

What it is: Picture connect is another forced association technique. It is designed to move idea generation away from traditional thinking and into areas ripe for fresh connections. This technique requires that you look at pictures and see what types of different connections you can make between the pictures and your problem or opportunity. You’d think simple pictures would work better; however, there is nothing scientific about such pictures, and any will do. What’s valuable is the process of forcing yourself to look at pictures which are unrelated to the challenge you are working on, then trying to draw connections to them.



One variation of this technique is to save interesting pictures from magazines to use the next time you face a problem or opportunity.

As usual, have someone in your group record your ideas on a flip chart.

Something it’s good for in software engineering: Proving to team members that they have to “de-focus” to see things that the act of bearing down on their work misses. The power of weak analogies and metaphors like these can be an eye-opening experience to someone who’s been working heads-down for a long time.

Technique – TRANSFORM

This is another brainstorming technique, for the top half of the 3-D diamond:

What it is: The TRANSFORM technique is a forced association technique using words to help you think about taking specific action relating to your problem or opportunity. Look at your challenge and see what connections come into your mind related to the different actions described by the words in the TRANSFORM chart below. Have someone from your team record the ideas on the flip chart.

T	Thicken	Transplant	Trim	Transfer
R	Reduce	Remove	Rearrange	Rotate
A	Adapt	Accelerate	Adopt	Add
N	Narrow	Naturalize	Neglect	Network
S	Substitute	Shift	Shrink	Simplify
F	Force	Flush	Fence	Frame
O	Oscillate	Off-Load	Organize	Optimize
R	Reverse	Recycle	Relax	Reform
M	Magnify	Maximize	Minimize	Move

Such material about permuting and combining ideas into new ones can be found in books like *Forks, Phonographs, and Hot Air Balloons: A Field Guide to Inventive Thinking*, by Robert J. Weber. Oxford University Press, 1992. ISBN 0-19-506402-X.

Something it's good for in software engineering: As a design exercise, it forces designers artificially to use a system they are not used to, an alternative to, say, normal object-oriented thinking. In this way, they can see relationships which might otherwise be buried. For example, notice the importance of verbs in this system, which would give added value to methods versus objects, in the way a designer considered a system.

Technique – Assumption Challenge

This is a *decision* technique, for the bottom half of the 3-D diamond:

What it is: When we work on any challenge or opportunity, we frequently make all kinds of assumptions that influence our ideas. The assumption challenge is designed to determine what the assumptions are, and see if by eliminating them we can generate ideas that will take us in new directions. Assumptions are built on our life experience. False assumptions are what often cause us to get stuck in the problem solving process. This technique will help you examine any assumptions your team is making about the challenge you are facing in your activity.

During brainstorming and decision making, everyone gets a basic picture of how they think the job has to be done, and as a result works in a more constrained problem-solving space than they really have to. We do this assumption-making on engineering projects, too. And in “real” markets, each competitor does this in a different way, depending on their corporate culture, existing products, and other forced aspects of their solutions. It is useful to consider how much of each organization’s problem-solving “box” is artificial, not really required for market success and perhaps working against that.

Something it's good for in software engineering: Here’s how assumption-challenge might work during decision-making, after you’ve done your main brainstorming session:

Step 1 – Use traditional brainstorming as a team to list any and all assumptions you are making about the opportunity presented in the problem or opportunity. These assumptions probably are built into the main brainstormed ideas you have listed. List the assumptions also on a flip chart. Do not discuss the list of assumptions at this time, just focus on generating the list.

If you can get your team successfully to write personal notes as they are brainstorming, you can get very rich results from that in this exercise. People will write down when they do not agree with a certain assumption, and they more likely will return to ask the group about that in step 2, below.

Step 2 – Now discuss the list of assumptions, and challenge your thinking to see if there are different ideas that arise from this new perspective. Record any new ideas on the flip chart.

This exercise may lead fairly quickly to the group’s deciding that one of the main brainstormed ideas looks much preferred over the others.

Technique – Rube Goldberg

This is another brainstorming technique, for the top half of the 3-D diamond:

What it is: The technique takes its name from artist and inventor Rube Goldberg, who drew humorous cartoons that appeared in newspapers around the world. They used combinations of everyday objects to accomplish simple tasks in a complex way. As an exercise for solving a software engineering problem, a Rube Goldberg is the

opposite of a simple, elegant solution. So, as in the Edsel method, this one can give you ideas about the right way or the parsimonious way, as you're doing it in this awful way.

Designing a "Rube Goldberg Machine" is now a rite-of-passage synthesis exercise for engineering students at many universities. "Rube Goldbergs," or "Rubies," are complicated, non-technical solutions to everyday problems. Usually the machine takes a number of distinct (and non-obvious) steps in completing the task.

Here's a description of one that Goldberg himself designed, taken from the WWW, where there's quite a bit of material on this subject. This famous "Rube Goldberg" was designed to open a garage door automatically (in the era before such things were common):

In this invention, one would drive the auto bumper against a mallet, pushing it down and exploding a cap, frightening a rabbit who runs toward his burrow, pulling string which discharges a pistol. The bullet penetrates a can from which the water drips into an aquarium. As the tide rises in the aquarium it elevates a floating cork upright, which pushes up the end of a see-saw, causing a flea to lose its balance and fall on a hound's tail. He wakes up and chases his tail round and round causing a platform to spin and turn on a faucet. Water runs through a hose, starting a revolving lawn sprinkler on which a rope winds itself, opening the garage door.

For extra credit, draw this gizmo from the words!

Something it's good for in software engineering: Making a "joke" version of the system, as complicated as possible, as a way of getting the designers to realize afterwards where their own unnecessary complications are.

More Tips

There are many resources with additional ideas for brainstorming, for creativity games, and for similar ways to stimulate new ideas. Some are targeted for teams to do together, while others work splendidly as a personal creativity aid.

Don't forget that all of these are methods that have been used successfully in problem solving, but that doesn't mean that they are guaranteed. Almost all of them assume that you've already tried the direct approach and failed, or don't want to because you want something new. Therefore, it's worthwhile trying a process with a chance of success which could hit big, or miss big.

Also, keep in mind that some of these games will be easier than others to get your colleagues engaged in. Most people feel that, in their group in business or industry, Idea Writing would be more immediately acceptable than Eyes Closed brainstorming, for instance. This does not mean that one is inherently more productive than the other, though. We'll leave you with that cognitive dissonance.

Some design or problem solving methods have very dedicated proponents, and it is probably useful to keep some perspective on any one technique. A couple examples of highly-touted methods which you could find helpful:

- **TRIZ**, whose enthusiastic advocates sometimes sound like they are claiming that it completely solves the problem of general engineering design. It has a lot of good features, of course. See <http://www.triz-journal.com/whatistriz.htm> for more information.
- **Synectics**, which extends brainstorming with many ancillary techniques and makes the process into a specific sequence of problem-solving steps. See <http://www.synecticsworld.com/> to contact them, or many web references describing their processes. Note that the step-by-step "Synectics process" as used in education is slightly different from the traditional Synectics brainstorming approach as it would be applied to SE problem solving.

Appendix -- Ford Facts...The Naming Of The Edsel

Just one of the mistakes-in-hindsight about this short-lived car was the process used to come up with its not-so-euphonious name --

In 1955, Ford's advertising agency Foote, Cone and Belding was charged with finding a suitable name for the soon to be released "E" car.

The agency had gathered no less than 18,000 suggestions ranging from "Resilient Bullet" to "Mongoose Civique" to the "Varsity Stroke." Unimpressed, Ford's Ernest Breech took delight in jokingly mispronouncing the ad agency's name as Foot, Corn And Bunion.

Meetings on the name selection were held everyday right after lunch in the new "E" divisions headquarters. The name suggestions were all printed in block letters, capitals only, six inches high, some white on black and some black on white

For a half hour, the names would be silently appear on a screen in front of an assembly of Ford executives. When a name pleased a particular viewer, they would yell "stop" and a discussion of the name would ensue.

David Wallace, who ran the meetings, became frustrated after several half hour sessions went by without a single "stop" being heard from the executives. To test their reactions, he inserted a slide bearing the letters B-U-I-C-K. When even this failed to draw a reaction from his viewers, he concluded that the sessions were being used as an opportunity for an after lunch nap.

The work progressed and the 18,000 names were reduced to 6,000 and then to 400. Nearly a year after starting, an ultimate list, on a single sheet of paper, was submitted on November 8th, 1956 to the Ford Executive Committee.

Of the 16 names it contained, consumer interviews favored "Corsair." Runner-up status went to "Ranger" and "Pacer." When Wallace told the executive committee that tests indicated "Corsair" received the highest level of consumer acceptance, the committee was unimpressed. They were not excited about any of the proposals.

Ernest Breech, who chaired the meeting in Henry II's absence, looked slowly at every member of the committee and declared that if this was the best selection available, "Why don't we just call it the Edsel!"

Edsel was the name that many of the newspapers had suggested when they first heard about the new "E" division that was announced in the summer of 1955. It seemed to fit well. An old family name on a brand new product.

When the idea to name the car the "Edsel" was presented to the Ford family they were not pleased at all. Henry II replied in the strictest word that his fathers name should not even be considered. And, when Charlie Moore, vice-president of public relations presented the idea Eleanor Ford, she slammed the door in his face.

Mrs. Edsel Ford was always a lady, especially in her dealings with Ford employees. But she was extremely unhappy about the use of her dead husbands name and so were her children.

It took an incredible amount of persuasiveness to convince the Ford family to overcome their instincts that the name "Edsel" was all wrong for the "E" car. In fact, the truth was stretched by Ford executives by overstating the Executive Committee's enthusiasm and public acceptance of the name.

The decision to name the "E" car the "Edsel" was a classic example of the growing isolation of Detroit from mainstream America. People in Detroit knew who Edsel Ford was and what he stood for but to the rest of the country "Edsel" was nothing more than a funny name. In fact, when market researchers asked people on the street to give their "immediate associations" upon hearing the name Edsel the came back with answers such as "Schmedsel", "Pretzel" and "Weasel." Nearly 40% simply reacted with "What??"

There are many fine sources of folklore (or better) about the Edsel. E.g., <http://en.wikipedia.org/wiki/Edsel>.