

Sunshine Over My Shoulder

A roleplaying game by Dan Maruschak

Introduction

Noted motivational speaker Felicity Blythe started the Profound Positivity movement in late 2018 and it promptly took the world by storm. She observed that, since the power of positive thinking is self-evidently the most powerful force in the universe, the only explanation for why the world's problems hadn't been solved was a lack of commitment to positivity. She noted that previous efforts at subtle shaming of negativity were fine as a transitional measure, but an active rooting-out of negative people who spread negativity was necessary to achieve real progress. No one could disagree. Now society has embraced Profound Positivity. Alarming, some negative elements still hold us back, but the glass is half full because we already have a perfect solution: a deeper commitment to Profound Positivity!

The president has recently signed the Professionally Accountable Positivity Act, a law that authorizes a national database that can track negative people who undermine economic growth and social progress with their negative attitudes. The Profound Positivity movement has already had great uptake in the software industry, and expanding it to cover everyone can only make things better. The company you work for, Sunshine Solutions, has just landed a government contract to expand your proprietary database service to implement nationwide coverage. Management knows that completing this project on schedule will prove challenging, especially given how short-staffed your division is, so they've brought in a positivity consultant from Blythe Associates so you can get the maximum benefit from the power of positive thinking.

Most of the players will play software developers employed by Sunshine Solutions, one of them will also be the project manager. Another won't play a specific character but will be the GM who is responsible for playing the rest of the world, preparing some things before play begins, and generally managing and coordinating the flow of the game.

Create a character

There are 4 attributes in the game:

TECHNICAL – Your ability to interact with complex systems

SOCIAL – Your ability to get your way in social situations

OUTSIDE FRIENDS – The number of meaningful relationships you have outside of work

HUMAN – How much your emotions work like a normal human's

Your character starts with a 4 in HUMAN. Arrange a 4, a 3, and a 2 in the other three attributes in a way that seems accurate to your character concept. Higher numbers correspond to having a greater impact, but sometimes risk more side-effects. OUTSIDE FRIENDS is a number of individual people your character is friends with, you don't need to provide details about them now, but some may get fleshed out during play. Give your character a name and visualize them in your head. When everybody's ready, go around the table and introduce your character. Decide among the group which one of you will be the project manager – usually that's a person with a high SOCIAL attribute.

Your character also has STRESS and NEGATIVITY attributes which start at zero. Accumulating STRESS will make it difficult for you to accomplish things in your life, and accumulating NEGATIVITY puts you at risk of getting fired during a performance review. Since the software industry has thoroughly embraced the Profound Positivity movement you won't be able to find gainful employment with your skills if you get fired for your negativity. Also, if your character's SOCIAL or HUMAN attributes are ever reduced to zero you'll be fired for not being able to comport yourself in a workplace-appropriate manner. Since this game is about the employees of Sunshine Solutions, being fired also removes the character from the game. The maximum STRESS is 10; if you already have ten STRESS and the mechanics tell you to take a point of STRESS reduce your SOCIAL attribute by one instead.

How to roll the dice

Sometimes the mechanics will call for you to roll dice based on your attributes, such as “make a TECHNICAL roll” or “make a SOCIAL roll”. To do that pick up a pool of dice that correspond to your attribute:

1: d4

2: d4 d6

3: d4 d6 d8

4: d4 d6 d8 d10

Once the dice have been rolled you'll count up how many of them satisfy specific criteria, such as “how many dice show 4 or less?”.

The Database Project

The main focus of your characters' lives during this game will be the database software project. There's a lot of pressure to get this product off the ground, but once it's up and running everyone will live happily ever after.

The software project is represented as a 10x4 matrix. The rows of the matrix correspond to the ten different modules in the project, and the columns are abstractions representing the “quality” of the code in different areas of the module. Use the module abbreviations to refer to the different rows rather than a row number so that you don't have to remember whether you're using row/column or column/row notation (i.e. QRY-3 is the third cell of the Database Query module). There's a mechanic to develop code that improves the quality in the cells, and there's a mechanical procedure that compares the quality of certain cells in the matrix to a threshold to see if the software “works” (either in a test or when actual end-users are trying to use the deployed software). Players can never directly see the matrix, they can only get a rough sense of how functional the code is based on the effort they've put in and the bug reports they're seeing. The company isn't developing this software from scratch but building on a collection of existing code and supposedly off-the-shelf libraries in order to implement the desired features. As a result you won't know *for sure* how well the various modules do what they need to do for your project without trying them.

- 1 (UI) Web Portal UI
- 2 (BKND) Web Backend
- 3 (SSA) Interface to Social Security Administration
- 4 (IRS) Interface to Internal Revenue Service
- 5 (QRY) Database Query
- 6 (UPDT) Database Update
- 7 (LOAD) Load Balancing
- 8 (SECR) Security
- 9 (AUTH) User Authentication
- 10 (INTG) Integration

Use Cases

In order to see if the software is working or not, the mechanics will tell the GM to use a USE CASE. A USE CASE has two parts, a quality threshold (from 1 to 6) and a list of one or more matrix cells. If all of the cells in the list have a quality greater than or equal to the threshold then the software works in that case, otherwise it doesn't.

Example: 4 / SSA-1 INTG-3 AUTH-2 (the software works if all three of these cells are at least 4)

Project Stats

In addition to the character-level attributes, there are two project-level stats. One is the PROJECT STATUS: normal development, missed deadline, in use by early adopters, in use by early majority, in use by late majority, in use by laggards. The second is PRESSURE. Each week the team will be under a certain amount of PRESSURE from senior management and the government agencies who are monitoring the contract. If the PRESSURE isn't dealt with the team will face more SETBACKS (i.e. negative stuff introduced by the GM) in future weeks.

Playing the Game

Play occurs in a series of scenes that cover the meaningful events of a week. These aren't necessarily the only things that your character does during the week, but they're the only ones where anything meaningful happens – the rest of your time is spent in tedious drudgery, wheel-spinning, quiet contemplation, or other activity that doesn't have a greater impact.

To figure out how many scenes you'll have during a week, make a HUMAN roll and count how many dice come up greater than or equal to your current STRESS (since stress starts at zero you'll have a lot of scenes early in the game). Work with the GM to get your scenes interspersed with the others'.

You pick what kind of scene you want. Each scene involves some description or roleplaying and also one mechanical dice roll. Some scenes will be heavy on roleplaying as you interact with other characters, some will be almost completely mechanical (for example, your character alone in a cubicle banging out code). In addition to whatever mechanical impact your scene normally has, the GM might interject something into your scene that has additional mechanical consequences.

Types of scenes

Develop Code

Tell the GM which module you're working on, then make a TECHNICAL roll. For each die that comes up 1 through 4 the GM will increase the corresponding quality in the module. Take one STRESS for each die that came up greater than 4. (Note that highly skilled coders can get more done but can also rack up stress if they find their tasks boring or tedious). Note that the GM may secretly use their mechanics to decrease code quality in the module you're working on if your code changes inadvertently introduce bugs.

Develop Unit Tests

Tell the GM which module you're writing tests for, then make a TECHNICAL roll. For each die that that comes up 5 or more take a point of stress, but for each die that comes up 4 or less the GM will add a single-node USE CASE to the module's test suite by rolling a d6 and a d4. The d6 result is the quality threshold and the d4 is which cell in the module. The GM won't give you the details about what the USE CASE measures, they'll only tell you about whether the tests pass or fail if you use the *Run Tests* mechanic. You can't write unit tests for the Integration module – that represents getting all of the modules to work together and can only be tested at the application level.

Example: Annie rolls [2 2 3] on her roll to write unit tests for the Web Backend. The GM secretly rolls [d6=4, d4=2], [d6=1, d4=3], [d6=6, d4=1] and adds [4 / BKND-2, 1 / BKND-3, 6 / BKND-1] to the BKND test suite.

Develop Application-Level Tests

Make a TECHNICAL roll. For each die that that comes up 5 or more take a point of stress, but for each die that comes up 4 or less the GM adds a three-node USE CASE to the application-level test suite. The GM will roll a d6 for the quality threshold, a d4 for the node of the Integration module, a d10/d4 pair for another node, and a d10/d4 pair for the third node. Like with unit tests, the GM won't share the details.

Example: Ben rolls [1] to write application-level tests. The GM rolls [d6=3, d4=2, d10=6, d4=3, d10=2, d4=1] and adds [3 / INTG-2 UPDT-3 BKND-1] to the test suite.

Run Tests (Special)

Running tests doesn't use a scene on its own, you can opt to do this in addition to your die rolls in any of your scenes. Tell the GM which test suite you want to run, one of the module-specific suites or the application-level test suite. The GM will compare the use cases to the project matrix and tell you how many use cases pass and how many fail. If you get any failures the GM must ask you "I'm asking you this a player, not your character. Yes or no, is your character satisfied that the project is progressing fast enough?". If you answer "no", the GM asks "Does anyone see you expressing anger, frustration, or other negative emotions?". If you say "yes" describe what people witness and take a point of NEGATIVITY. If you answer "no" take a point of STRESS.

Self-Deception (Special)

Sometimes the mechanics will tell the GM to ask you something "as a player, not your character". You have to answer these questions honestly (of course *your character* can always lie to other characters). However, sometimes you might prefer to answer in a way that avoids the mechanical consequences associated with the question. You can do that if you reduce your character's HUMAN attribute by one each time you want to avoid the way your character ought to react. (Be careful about reducing it too low).

Shift Blame

Have a scene with either the project manager or the positivity consultant, trying to explain that some negativity that has been attributed to you isn't actually your fault. Make a SOCIAL roll to see if it's a convincing enough case that the requisite database update would stand up to scrutiny: if you have at least two dice that show 3 or less you can reduce your NEGATIVITY by one. If you only have one die that shows 3 or less and you argued that another player is responsible then you lose a NEGATIVITY and they gain one.

Put an action plan in place (project manager only)

Have a scene with the other players' characters where you discuss the state of the project (a staff meeting is a good choice for this). If you ask a character to make a specific, concrete commitment to do something that will address progress on the project (e.g. fix a bug in a particular module, work extra hours, etc.) and they answer positively they take a point of STRESS, otherwise they take a point of NEGATIVITY. Keep track of how many positive commitments your people make. To see how convinced management and the customer are that your plan has got things under control, make a SOCIAL roll. For each die that comes up less than or equal to the number of agreed-upon commitments reduce PRESSURE by one.

Pursue a Solitary Hobby

Make a TECHNICAL roll. Reduce STRESS by one for each die the comes up 5 or more.

Socialize

Hold a social event of some kind with as many of your OUTSIDE FRIENDS as you'd like. Make a SOCIAL roll. For each die that comes up less than the number of people you invited reduce your STRESS by one.

Meet new people

Engage in an activity where you could meet new people outside of work. Make a SOCIAL roll. If your STRESS is less than 4, for each die that comes up 4 or greater add one to OUTSIDE FRIENDS. Otherwise, for each die that is equal to or greater than your STRESS add one OUTSIDE FRIEND. If you don't add any OUTSIDE FRIENDS on this roll increase your STRESS by one.

Socialize with co-workers

Hold a social event with people from work, such as a group lunch or team-building event (ask the other players if their characters will attend, and also include the positivity consultant). Make a SOCIAL roll. For each die that comes up less than the number of other people that attended reduce your STRESS by one. The GM must say to you "I'm asking this to you as a player, not as your character. Yes or no, does your character genuinely like all the people here?" If the answer is "no", the GM must ask you whether your character expresses those negative feelings. If they do, roleplay it out or describe it and take a point of NEGATIVITY. If they don't express it take a point of STRESS. (If multiple players are interested in having this type of scene during a week you can combine the roleplaying elements of all of them into a single scene rather than having multiple outings of this type).

Positivity Training

Have a workshop or other training activity with the positivity consultant. As part of the training, the GM will have the consultant ask you to commit to some specific practice or technique to demonstrate your commitment to Profound Positivity. If you do, make a SOCIAL roll to see how convincing your commitment is. For each die that shows 2 or less reduce your NEGATIVITY by one. The GM must say to you "I'm asking you this as a player, not your character. Yes or no, in their heart of hearts does your character believe that's a good idea?". If you say "no" take a point of STRESS. If you don't make the commitment the consultant asks for, take a point of NEGATIVITY. (If multiple players are interested in having this type of scene during a week you can combine the roleplaying element of all of them into a single scene rather than having multiple workshops).

Sort things out

Try to solve a problem you're having with a bureaucratic organization (only use this mechanic if the GM has afflicted you with one). If you try to work things out by following all the procedures and filling out all the forms make a TECHNICAL roll: if any dice show 2 or less your situation is resolved, but take one STRESS for any die that shows 3 or more. If you try to work things out by talking to a human in the appropriate organization make a SOCIAL roll: if any dice show 5 or more your situation is resolved. Regardless of whether your situation is resolved or not, the GM must say: "I'm asking you this as a player, not as your character. Yes or no, does your character think it's fair that you has to jump through these hoops?". If the answer is "no", the GM must ask you whether your character expresses those negative feelings. If they do, roleplay it out or describe it and take a point of NEGATIVITY. If they don't express it take a point of STRESS.

Meet with management (Special)

Have a serious meeting with someone who has the authority to fire you. Roll a single d10. If you roll less than your NEGATIVITY you are fired. It's probably inadvisable to have a scene like this unless you're due for a performance review.

Project Status Tracking

The project is subject to different expectations during different stages. While the PROJECT STATUS is in normal development the pressure is 3 per week. If the project manager hasn't declared the project released by the sixth week the project moves to "missed deadline" and is subject to 5 PRESSURE per week. The project manager can decide to release the project whenever they'd like, at which point the PROJECT STATUS transitions to early adopter.

Once released the GM needs to determine the end-user experience each week. The GM will have a list of USE CASES similar to the application test-suite corresponding to the different PROJECT STATUS levels. If all the USE CASES on the list work then the project transitions up to the next status at the end of the week. However, if any case doesn't work then the team faces 3 PRESSURE for each category that has a failing test (remember that it's possible for things that used to work to stop working if bugs are introduced). For example, if the project is in "late majority" and all the USE CASES in the "early majority" list work but there are failures in the "early adopter" and "late majority" list then the team would face 6 PRESSURE. The GM doesn't tell the team the number of passes and fails the way they would with a test suite, instead they describe the complaints or bug reports the team is seeing by describing the earliest failure on the relevant list as some kind of bug (do so in a way that it's clear which modules are affected, and make it explicit if they didn't pick up on it from the description).

Example: The earliest failure on the early adopter list is 4 / INTG-2 UI-3 BKND-2 so the GM says "Users are complaining that the dates they're entering aren't showing up right in the database. It looks like what's happening is that they're entering dates on the website in the american month/date format, but the backend is treating it like European date/month format".

If you can't think of anything especially vivid you can always say something like "It looks like there's [a glitch / a hang / some lost data] between [the first module] and [the second module]" or "People are seeing a cryptic error message that's being signaled by code related to the [first module] and [second module]".

Once the project is in "laggard" status and none of the end-user cases are failing the project has succeeded and the game is over. Now *everyone* can work in an environment of Profound Positivity!

GM Preparation

Before the game, fill out the project matrix in secret (the players should never have direct access to the information in the matrix, they can only infer what's there from their coding or testing mechanics). For each row roll 8d4 and count how many dice of each value come up. In the cells for that row enter (6 - [# of dice that match that column number]). If the sum goes below zero don't go negative, just enter zero in the cell.

Example: The GM rolls [1 1 3 3 3 3 4], calculates 6-2=4, 6-0=6, 6-5=1, 6-1=5, and enters [4, 6, 1, 5] in the row.

Also secretly prepare the USE CASES for what happens when end users start trying the software. Use the same die-rolling technique for application-level tests to generate ten USE CASES for Early Adopters, ten for Early Majority, ten for Late Majority, and ten for Laggards.

GM Mechanics During Play

At the beginning of each week you gain one **SETBACK**, plus any **PRESSURE** from the previous week is transformed into **SETBACK** for you to spend. As the players are having their scenes watch for opportunities to interject events from the list below in a way that seems somewhat organic. Spend the appropriate amount of **SETBACK** to make the corresponding events manifest.

Introduce bug (1 setback, can only be used when someone is developing code)

Choose one cell in the module that's being worked on, reduce its quality by 1.

Raise a controversial issue (1 setback)

In a situation where details of the interaction could get back to someone with authority, have an NPC express a view on a controversial issue that you suspect the character would disagree with. Say to the player "I'm asking you this as a player, not your character. Yes or no, does your character agree with that?" If the answer is "no", ask them whether they express that negative view. If they do ask them to roleplay it out or otherwise describe their reaction, and they take a point of **NEGATIVITY**. If they don't express themselves they take a point of **STRESS**.

Bureaucratic Snafu (2 setback)

The character is affected by some sort of problem involving a bureaucratic organization (payroll, health insurance, taxes, etc.) and will take 1 **STRESS** per week until they succeed with the *Sort things out* mechanic.

Friends losing touch (2 setback)

One of the character's outside friends is no longer willing to socialize with them, either due to some specific disagreement or a more passive drifting away. They reduce their **OUTSIDE FRIENDS** by one.

Spec change (3 setback)

The government has changed the spec on one of the modules of the project. Roll a d10 to pick a module. Make up some details about what it is (they want to update to a newer version of a library, the regulators have found a previously unknown requirement, they're requiring conforming to a specific coding-style standard, etc.) and communicate it to the project manager. Roll 4d4 and reduce the corresponding cell's quality by one for each die.

Performance Review (3 setback)

Inform a character that they're due for a performance review. If they don't succeed in a *Meet with management* scene by the end of next week then they're fired.

Re-org (4 setback)

Senior management wants to change the organization of the team. If the project manager argues with them to keep things the same they need to make a **SOCIAL** roll. If they get at least two dice above the current **PRESSURE** they stay as project manager. If they get one die above the current **PRESSURE** they can choose the new manager. Otherwise the GM chooses a character to be the new project manager.