```
import pytest
from datetime import datetime
from swarms.artifacts.main_artifact import Artifact, FileVersion


def test_file_version():
    version = FileVersion(
        version_number=1,
        content="Initial content",
        timestamp=datetime.now(),
    )
    assert version.version_number == 1
    assert version.content == "Initial content"


def test_artifact_creation():
    artifact = Artifact(file_path="test.txt", file_type=".txt")
    assert artifact.file_path == "test.txt"
    assert artifact.file_type == ".txt"
    assert artifact.contents == ""
    assert artifact.versions == []
    assert artifact.edit_count == 0


def test_artifact_create():
    artifact = Artifact(file_path="test.txt", file_type=".txt")
    artifact.create("Initial content")
    assert artifact.contents == "Initial content"
    assert len(artifact.versions) == 1
    assert artifact.versions[0].content == "Initial content"
    assert artifact.edit_count == 0


def test_artifact_edit():
    artifact = Artifact(file_path="test.txt", file_type=".txt")
    artifact.create("Initial content")
    artifact.edit("First edit")
    assert artifact.contents == "First edit"
    assert len(artifact.versions) == 2
    assert artifact.versions[1].content == "First edit"
    assert artifact.edit_count == 1


def test_artifact_get_version():
    artifact = Artifact(file_path="test.txt", file_type=".txt")
    artifact.create("Initial content")
    artifact.edit("First edit")
    version = artifact.get_version(1)
    assert version.content == "Initial content"


def test_artifact_get_contents():
    artifact = Artifact(file_path="test.txt", file_type=".txt")
    artifact.create("Initial content")
    assert artifact.get_contents() == "Initial content"


def test_artifact_get_version_history():
    artifact = Artifact(file_path="test.txt", file_type=".txt")
    artifact.create("Initial content")
    artifact.edit("First edit")
    history = artifact.get_version_history()
    assert "Version 1" in history
    assert "Version 2" in history


def test_artifact_to_dict():
    artifact = Artifact(file_path="test.txt", file_type=".txt")
    artifact.create("Initial content")
    artifact_dict = artifact.to_dict()
    assert artifact_dict["file_path"] == "test.txt"
    assert artifact_dict["file_type"] == ".txt"
    assert artifact_dict["contents"] == "Initial content"
    assert artifact_dict["edit_count"] == 0


def test_artifact_from_dict():
    artifact_dict = {
        "file_path": "test.txt",
        "file_type": ".txt",
        "contents": "Initial content",
        "versions": [
            {
                "version_number": 1,
                "content": "Initial content",
                "timestamp": datetime.now().isoformat(),
            }
        ],
        "edit_count": 0,
    }
    artifact = Artifact.from_dict(artifact_dict)
    assert artifact.file_path == "test.txt"
    assert artifact.file_type == ".txt"
    assert artifact.contents == "Initial content"
    assert artifact.versions[0].content == "Initial content"
    assert artifact.edit_count == 0


# Run the tests
if __name__ == "__main__":
    pytest.main()
```