

# Тестовое задание, анализ механики гильдий в MMO RPG.

Для начала хотелось бы подробнее разобрать механику гильдий.

Преимущества гильдии:

- Гильдии увеличивают количество контента, доступного для игрока (участие в ГВ, рейдах, выполнений квестов гильдии, поручений).
- Таланты/бафы гильдии, усиливающие игроков и способствующие более приятному игровому опыту.
- Уникальные предметы из магазина гильдий.
- Сплочение игроков, где новичок не останется без поддержки, а более опытные товарищи найдут для себя контент в виде сопровождения новичков.

Для жанра MMO RPG данная механика является чуть ли не основополагающей, предоставляя игрокам возможность соревноваться между гильдиями. Это дополнительная мотивация для игрока оставаться в проекте, а для некоторых пользователей гильдия является ключевым элементом в игровом опыте.

“Порог входа” для открытия механики - 8 уровень персонажа и небольшое количество внутриигровой валюты. Этот порог низок, поэтому создать гильдию может любой игрок.

Говоря языком продуктовой аналитики, данная механика рассчитана на увеличение retention (удержания) игроков в игре, независимо от их наигрыша в проект. Соответственно увеличивается lifetime (время жизни пользователя в игре), что в совокупности с paying share (доля платящих) и ARPPU (средний чек платящего пользователя) приводит к увеличению важнейшей метрики бизнеса LTV (выручка с пользователя за время его жизни в проекте).

Несмотря на простоту механики, есть множество факторов, требующих анализа. Рассмотрим по порядку гипотезы, выдвинутые коллегой геймдизом.

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
import seaborn as sns
sns.set(rc={'figure.figsize': (14,8)})
```

## 1. Провожу "распаковку" данных

```
In [2]: energy = pd.read_csv('C:/Users/123/Desktop/Игровая аналитика/energy.csv')
```

```
In [3]: guild_data = pd.read_csv('C:/Users/123/Desktop/Игровая аналитика/guild_data.csv')
```

```
In [4]: payers = pd.read_csv('C:/Users/123/Desktop/Игровая аналитика/payers.csv')
```

Сразу же с ними знакомлюсь, правильно ли все отображается

```
In [5]: guild_data.head() # также смотрел и .tail
```

```
Out[5]:    datetime  creator_character_id  guild_id  n_added_characters  n_removed_characters  n_talents  level  n_missions
```

0	2022-12-12 23:13:42	15092203	78698	1	1.0	NaN	NaN
1	2022-12-12 22:29:09	15961764	78604	2	2.0	NaN	NaN
2	2022-12-12 21:01:53	16187808	78510	27	19.0	9.0	2.0
3	2022-12-12 00:16:52	21618959	77899	42	35.0	12.0	3.0
4	2022-12-12 22:13:50	23341705	78557	6	6.0	4.0	2.0

In [6]: `payers.head()` # также смотрел и `.tail`

Out[6]:

	revenue	character_id
0	102.897877	13642352
1	105.839569	29703528
2	50.040612	28081545
3	18.044814	12663426
4	5.913989	16071934

In [7]: `energy.head()` # также смотрел и `.tail`

Out[7]:

	energy_delta	context_type	guild_id
0	1148686	GuildMissionRewardContext	90166
1	1018116	GuildMissionRewardContext	81283
2	907440	GuildMissionRewardContext	79262
3	889109	GuildMissionRewardContext	84667
4	858954	GuildMissionRewardContext	77241

## 2. Провожу первичный анализ данных

In [8]: `guild_data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 965 entries, 0 to 964
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   datetime                             965 non-null   object
1   creator_character_id                 965 non-null   int64
2   guild_id                             965 non-null   int64
3   n_added_characters                   965 non-null   int64
4   n_removed_characters                 866 non-null   float64
5   n_talents                            672 non-null   float64
```

```
6   level          572 non-null    float64
7   n_missions_completed  895 non-null    float64
8   n_times_flag_turned_on  81 non-null     float64
9   mean_place        59 non-null     float64
10  mean_rating        59 non-null     float64
11  n_seasons         59 non-null     float64
dtypes: float64(8), int64(3), object(1)
memory usage: 90.6+ KB
```

```
In [9]: guild_data.describe()
```

	creator_character_id	guild_id	n_added_characters	n_removed_characters	n_talents	level	n_mi
count	9.650000e+02	965.000000	965.000000	866.000000	672.000000	572.000000	
mean	2.323599e+07	72964.000000	32.888083	31.047344	15.422619	3.097902	
std	3.926890e+06	13099.642839	36.893785	34.651878	15.254326	1.176624	
min	1.063135e+07	50310.000000	1.000000	1.000000	1.000000	2.000000	
25%	2.114097e+07	61637.000000	4.000000	5.000000	6.000000	2.000000	
50%	2.395866e+07	72964.000000	20.000000	19.000000	11.000000	3.000000	
75%	2.631834e+07	84291.000000	49.000000	47.000000	20.000000	4.000000	
max	2.909053e+07	95618.000000	283.000000	265.000000	109.000000	8.000000	

Данные по гильдиям выглядят хорошо. Нужно убедиться, что все айдишники гильдий уникальны в выгрузке

```
In [10]: guild_data.guild_id.duplicated().sum() # проверяем на дубли
```

```
Out[10]: 0
```

```
In [11]: guild_data.guild_id.nunique() # считаем число уникальных
```

```
Out[11]: 965
```

Супер, все сходится, двигаемся дальше

```
In [12]: payers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20430 entries, 0 to 20429
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype  
---  -
 0   revenue         20420 non-null  float64
 1   character_id    20430 non-null  int64  
dtypes: float64(1), int64(1)
memory usage: 319.3 KB
```

```
In [13]: payers.describe()
```

	revenue	character_id
count	20420.000000	2.043000e+04
mean	31.233120	2.767138e+07

<b>std</b>	54.885478	8.226633e+06
<b>min</b>	0.000000	1.062438e+07
<b>25%</b>	5.510835	2.089467e+07
<b>50%</b>	10.285234	2.826496e+07
<b>75%</b>	30.633660	3.452353e+07
<b>max</b>	499.498653	4.186029e+07

In [14]: `payers.character_id.nunique()` *# проверяем правильность выгрузки данных*

Out[14]: 20430

Судя по минимальной выручке, здесь данные по всем пользователям, а не только платящим. Хотелось бы уточнить это у коллег, но пока поищу ответ на этот вопрос по-своему.

In [15]: `len(payers[payers.revenue == 0])`

Out[15]: 25

Всего 25 персонажей обошлись без доната! Если это все пользователи за период, то paying share проекта просто потрясающий :) Посмотрим ARPPU без учета попавших в выборку 25 неплатящих.

In [16]: `payers[payers.revenue > 0].revenue.mean()`

Out[16]: 31.271404968659255

Конечно, я подозреваю, что неплатящие пользователи случайно попали в выборку (или не случайно, если они являются ГМами гильдий(: )

## Немного поиграли с данными, двигаемся дальше.

In [17]: `energy.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 971 entries, 0 to 970
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   energy_delta  971 non-null    int64
 1   context_type  971 non-null    object
 2   guild_id     971 non-null    int64
dtypes: int64(2), object(1)
memory usage: 22.9+ KB
```

In [18]: `energy.describe()`

Out[18]:

	energy_delta	guild_id
<b>count</b>	9.710000e+02	971.000000
<b>mean</b>	7.319469e+04	72661.476828

<b>std</b>	1.428501e+05	12932.582339
<b>min</b>	5.000000e+01	50310.000000
<b>25%</b>	4.200000e+03	61613.500000
<b>50%</b>	2.248000e+04	72353.000000
<b>75%</b>	7.136000e+04	84009.000000
<b>max</b>	1.148686e+06	95618.000000

In [19]: `energy.guild_id.nunique()`

Out[19]: 841

In [20]: `energy.context_type.unique()`

Out[20]: `array(['GuildMissionRewardContext', 'CompleteGuildRaidMatchContext',  
'ConvertKnowledgeToGuildEnergyContext'], dtype=object)`

Судя по количеству уникальных гильдий, не все получали энергию. Также, мы имеем 3 типа источников энергии. В целом, данные также выглядят здоровыми.

## 3. Создание общего фрейма данных

Для удобства и наглядности, хочется объединить все данные в один датафрейм (таблицу). Правда, перед этим, данные предстоит немного предобработать.

In [21]: `# переименуем столбцы в таблице с платежами, чтобы можно было объединить с данными о гил  
payers.columns = ['revenue', 'creator_character_id']`

In [22]: `full_data = guild_data.merge(payers, on='creator_character_id', how='left')`

In [23]: `full_data`

Out[23]:

	<b>datetime</b>	<b>creator_character_id</b>	<b>guild_id</b>	<b>n_added_characters</b>	<b>n_removed_characters</b>	<b>n_talents</b>	<b>level</b>	<b>n_missio</b>
--	-----------------	-----------------------------	-----------------	---------------------------	-----------------------------	------------------	--------------	-----------------

<b>0</b>	2022-12-12 23:13:42	15092203	78698	1	1.0	NaN	NaN	
----------	---------------------	----------	-------	---	-----	-----	-----	--

<b>1</b>	2022-12-12 22:29:09	15961764	78604	2	2.0	NaN	NaN	
----------	---------------------	----------	-------	---	-----	-----	-----	--

<b>2</b>	2022-12-12 21:01:53	16187808	78510	27	19.0	9.0	2.0	
----------	---------------------	----------	-------	----	------	-----	-----	--

<b>3</b>	2022-12-12 00:16:52	21618959	77899	42	35.0	12.0	3.0	
----------	---------------------	----------	-------	----	------	------	-----	--

<b>4</b>	2022-12-12 22:13:50	23341705	78557	6	6.0	4.0	2.0	
----------	---------------------	----------	-------	---	-----	-----	-----	--

...	...	...	...	...	...	...	...
<b>960</b>	2022-11-10 13:18:37	20594240	54493	16	7.0	11.0	3.0
<b>961</b>	2022-11-10 16:18:24	20755700	54587	35	28.0	14.0	3.0
<b>962</b>	2022-11-10 00:36:45	21310943	54258	14	6.0	3.0	NaN
<b>963</b>	2022-11-10 01:03:53	21778372	54305	6	1.0	8.0	2.0
<b>964</b>	2022-11-10 06:56:10	21945697	54446	41	35.0	14.0	3.0

965 rows × 13 columns

Теперь мы можем видеть какое количество реальных денег вкладывают главы различных гильдий. Далее, нам нужно добавить столбец с получением энергии. Для этого нужно сгруппировать по гильдиям таблицу *energy* и просуммировать всю полученную из различных источников энергию.

In [24]: `energy_new = pd.DataFrame(data = energy.groupby('guild_id')['energy_delta'].sum().reset_`

In [25]: `energy_new.columns = ['guild_id', 'total_energy']`

In [26]: `energy_new`

Out[26]:

	guild_id	total_energy
<b>0</b>	50310	1960
<b>1</b>	50357	94280
<b>2</b>	50404	24680
<b>3</b>	50451	6440
<b>4</b>	50545	27060
...	...	...
<b>836</b>	95383	560
<b>837</b>	95477	94540
<b>838</b>	95524	73200
<b>839</b>	95571	26540
<b>840</b>	95618	16580

841 rows × 2 columns

По-моему, получилось неплохо :) Соединим теперь эти данные с

## общим датафреймом

```
In [27]: full_data = full_data.merge(energy_new, on='guild_id', how='left')
```

```
In [28]: full_data.head(10)
```

```
Out[28]:
```

	datetime	creator_character_id	guild_id	n_added_characters	n_removed_characters	n_talents	level	n_missions
0	2022-12-12 23:13:42	15092203	78698	1	1.0	NaN	NaN	
1	2022-12-12 22:29:09	15961764	78604	2	2.0	NaN	NaN	
2	2022-12-12 21:01:53	16187808	78510	27	19.0	9.0	2.0	
3	2022-12-12 00:16:52	21618959	77899	42	35.0	12.0	3.0	
4	2022-12-12 22:13:50	23341705	78557	6	6.0	4.0	2.0	
5	2022-12-12 00:26:46	23748506	77946	4	3.0	4.0	2.0	
6	2022-12-12 19:43:20	23909299	78369	94	83.0	24.0	4.0	
7	2022-12-12 00:07:19	24425304	77805	27	15.0	5.0	2.0	
8	2022-12-12 20:06:29	25081103	78416	1	1.0	NaN	NaN	
9	2022-12-12 15:29:23	25567139	78228	11	11.0	NaN	NaN	

Избавимся от пропусков, заменив их значением "0" для удобного проведения расчётов

```
In [29]: full_data = full_data.fillna(0)
```

```
In [30]: full_data.total_energy.median()
```

```
Out[30]: 16580.0
```

Супер, все необходимые данные собраны вместе. Теперь можно начинать выполнение самого задания :)

# 4. Проверка первой гипотезы.

**Нулевая гипотеза:** Большинство создаваемых гильдий не активны.

**Альтернативная гипотеза:** Большинство создаваемых гильдий активны.

Для начала, нужно определить какая гильдия является активной, а какая нет. Гильдия может быть активна даже если в ее составе совсем немного людей, но их объединяют общие цели и интересы. Поэтому нужно выделить определяющие факторы "неактивной" гильдии.

Возьмем такие факторы:

- 1. Гильдия, для которой в анализируемый период количество приглашенных участников = количество исключенных участников, считается не активной.
- 2. Гильдия, которая в анализируемый период не повысила свой уровень ни разу, считается не активной.
- 3. Гильдия, которая за анализируемый период получила менее 15 000\* энергии, считается не активной.

15 000\* - показатель выбран экспертно, отталкиваясь от медианы общих данных по энергии

```
In [31]: full_data['n_removed_characters'] = full_data['n_removed_characters'].astype(int)
full_data['level'] = full_data['level'].astype(int)
full_data['total_energy'] = full_data['total_energy'].astype(int)
```

```
In [32]: # добавляем столбец, в котором будет видно количество участников гильдии по итогам анали
full_data['characters'] = full_data['n_added_characters'] - full_data['n_removed_charact
```

```
In [33]: full_data
```

Out[33]:

	datetime	creator_character_id	guild_id	n_added_characters	n_removed_characters	n_talents	level	n_mission
0	2022-12-12 23:13:42	15092203	78698	1	1	0.0	0	
1	2022-12-12 22:29:09	15961764	78604	2	2	0.0	0	
2	2022-12-12 21:01:53	16187808	78510	27	19	9.0	2	
3	2022-12-12 00:16:52	21618959	77899	42	35	12.0	3	
4	2022-12-12 22:13:50	23341705	78557	6	6	4.0	2	
...	...	...	...	...	...	...	...	...
960	2022-11-10 13:18:37	20594240	54493	16	7	11.0	3	
961	2022-11-10	20755700	54587	35	28	14.0	3	



	16:18:24						
962	2022-11-10 00:36:45	21310943	54258	14	6	3.0	0
963	2022-11-10 01:03:53	21778372	54305	6	1	8.0	2
964	2022-11-10 06:56:10	21945697	54446	41	35	14.0	3

965 rows × 15 columns

Теперь по установленным критериям мы можем получить число гильдий, которые стали неактивными в течение анализируемого периода.

```
In [34]: full_data['total_energy'].sort_values().head(450)
```

```
Out[34]: 0          0
509         0
507         0
111         0
499         0
...
501      13320
85       13420
287      13420
679      13480
368      13500
Name: total_energy, Length: 450, dtype: int32
```

```
In [35]: len(full_data[(full_data['level'] == 0) &
                        (full_data['total_energy'] < 15000)])
```

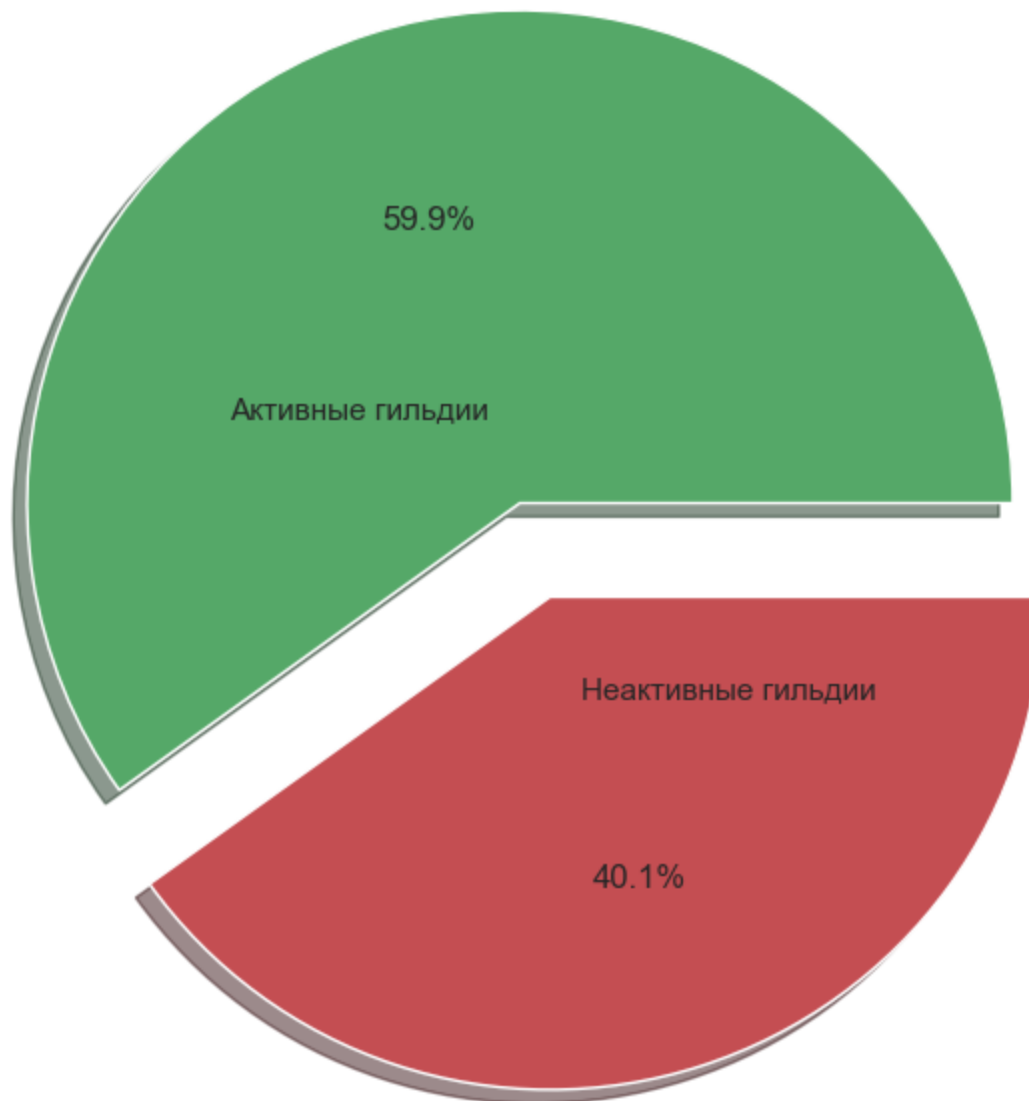
```
Out[35]: 387
```

```
In [36]: print('Результат в доле от общего числа: ' f"{387/965:.1%}")
```

```
Результат в доле от общего числа: 40.1%
```

## Визуализируем результат

```
In [37]: plt.pie([965-387, 387], labels=['Активные гильдии', 'Неактивные гильдии'], colors=['g', 'r'],
              radius=1, shadow=True, explode=[0.1, 0.1], labeldistance=0.2);
```



По результатам проведенного исследования, мы можем опровергнуть нулевую гипотезу. Большинство создаваемых гильдий активны.

## 5. Проверка второй гипотезы.

**Нулевая гипотеза:** Игроки, покупающие премиум товары, создают более активные гильдии, поскольку, инвестируя в игру, пользователь больше ценит вложенные усилия.

**Альтернативная гипотеза:** Активность гильдий не зависит от количества купленных ГМ (Guild Master) премиум товаров.

```
In [38]: full_data.head(5)
```

```
Out[38]:
```

	datetime	creator_character_id	guild_id	n_added_characters	n_removed_characters	n_talents	level	n_missions
0	2022-12-12	15092203	78698	1	1	0.0	0	

23:13:42

1	2022-12-12 22:29:09	15961764	78604	2	2	0.0	0
2	2022-12-12 21:01:53	16187808	78510	27	19	9.0	2
3	2022-12-12 00:16:52	21618959	77899	42	35	12.0	3
4	2022-12-12 22:13:50	23341705	78557	6	6	4.0	2

Посмотрим на топ 100 самых активных гильдий.

Определяющими факторами активности будем считать:

- суммарное количество изученных талантов
- максимально достигнутый уровень гильдии
- суммарное количество выполненных кв гильдий
- суммарное полученное количество энергии
- суммарное количество сезонов войн, в которых было принято участие

Здесь мы берем только абсолютные показатели, которые не будут искажать анализ. Такие показатели как "среднее место в ГВ" и тд не является показательным, так как может зависеть от многих факторов и не так определяет успешность ГИ как ее активность.

```
In [39]: top100_guilds = full_data.sort_values(by=['n_talents', 'level', 'n_missions_completed',  
                                              ascending=False).reset_index(drop=True).head(100)
```

Оставим только данные, необходимые для анализа

```
In [40]: top100_guilds = top100_guilds[['guild_id', 'n_talents', 'level', 'n_missions_completed',
```

```
In [41]: top100_guilds.head(10)
```

```
Out[41]:
```

	guild_id	n_talents	level	n_missions_completed	n_seasons	revenue	total_energy	characters
0	90166	109.0	8	279.0	0.0	0.000000	1847185	21
1	84667	103.0	8	278.0	0.0	5.926916	1746857	22
2	81283	98.0	8	267.0	0.0	0.000000	1748236	21
3	74092	92.0	8	265.0	0.0	0.000000	1651500	23
4	71836	85.0	7	254.0	0.0	13.710839	1379963	22
5	93785	83.0	7	268.0	0.0	125.444119	1371374	21
6	77241	80.0	7	251.0	0.0	190.773549	1432603	22
7	77147	79.0	7	270.0	0.0	192.733336	1271578	23
8	57031	76.0	6	230.0	0.0	52.466201	791150	20
9	73669	75.0	7	271.0	0.0	0.000000	1389845	19

В целых трёх из ТОП-4 гильдий ГМы обходятся без покупок премиум товаров! Молодцы, такие лидеры не могут не вызывать приятные эмоции! :)

Теперь сортируем данный топ в порядке убывания суммы покупок ГМов, сохраняя столбец с их местом в топе.

```
In [42]: top100_donations = top100_guilds.sort_values(by='revenue', ascending=False).reset_index()
```

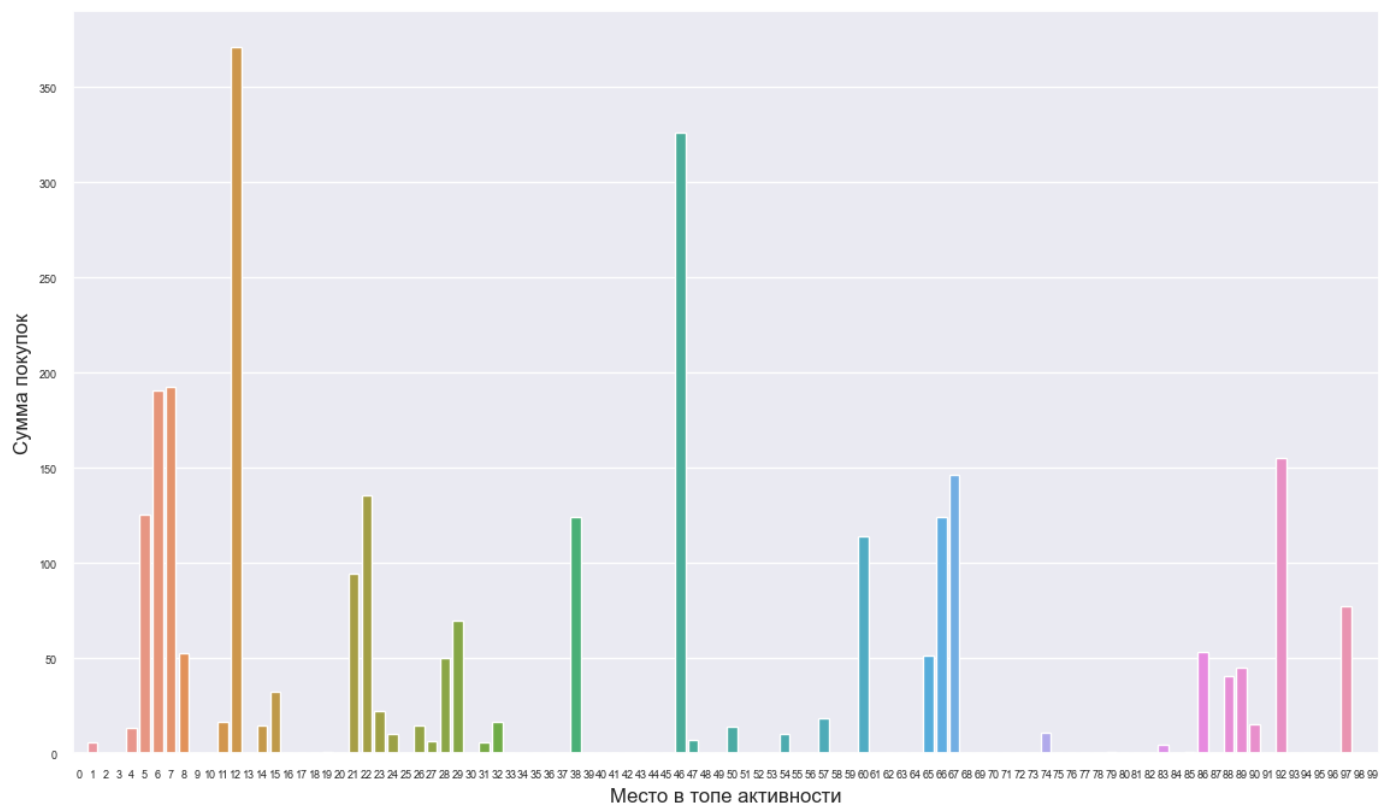
```
In [43]: top100_donations.head()
```

```
Out[43]:
```

	index	guild_id	n_talents	level	n_missions_completed	n_seasons	revenue	total_energy	characters
0	12	79262	71.0	7	266.0	0.0	371.374727	1298954	21
1	46	91670	36.0	5	239.0	3.0	326.440396	376100	14
2	7	77147	79.0	7	270.0	0.0	192.733336	1271578	23
3	6	77241	80.0	7	251.0	0.0	190.773549	1432603	22
4	92	68593	27.0	4	218.0	0.0	155.066612	197820	13

Визуализируем полученные данные, чтобы составить полную картину.

```
In [44]: sns.barplot(data=top100_donations, x='index', y="revenue")
plt.xlabel('Место в топе активности')
plt.ylabel('Сумма покупок');
plt.tick_params (axis='both', which='major', labelsize= 6.5)
```



Визуально мы видим увеличение концентрации платящих на более высоких позициях топа. Теперь будем сужать рассматриваемый топ чтобы отследить концентрацию платящих ГМ ближе к верхушке.

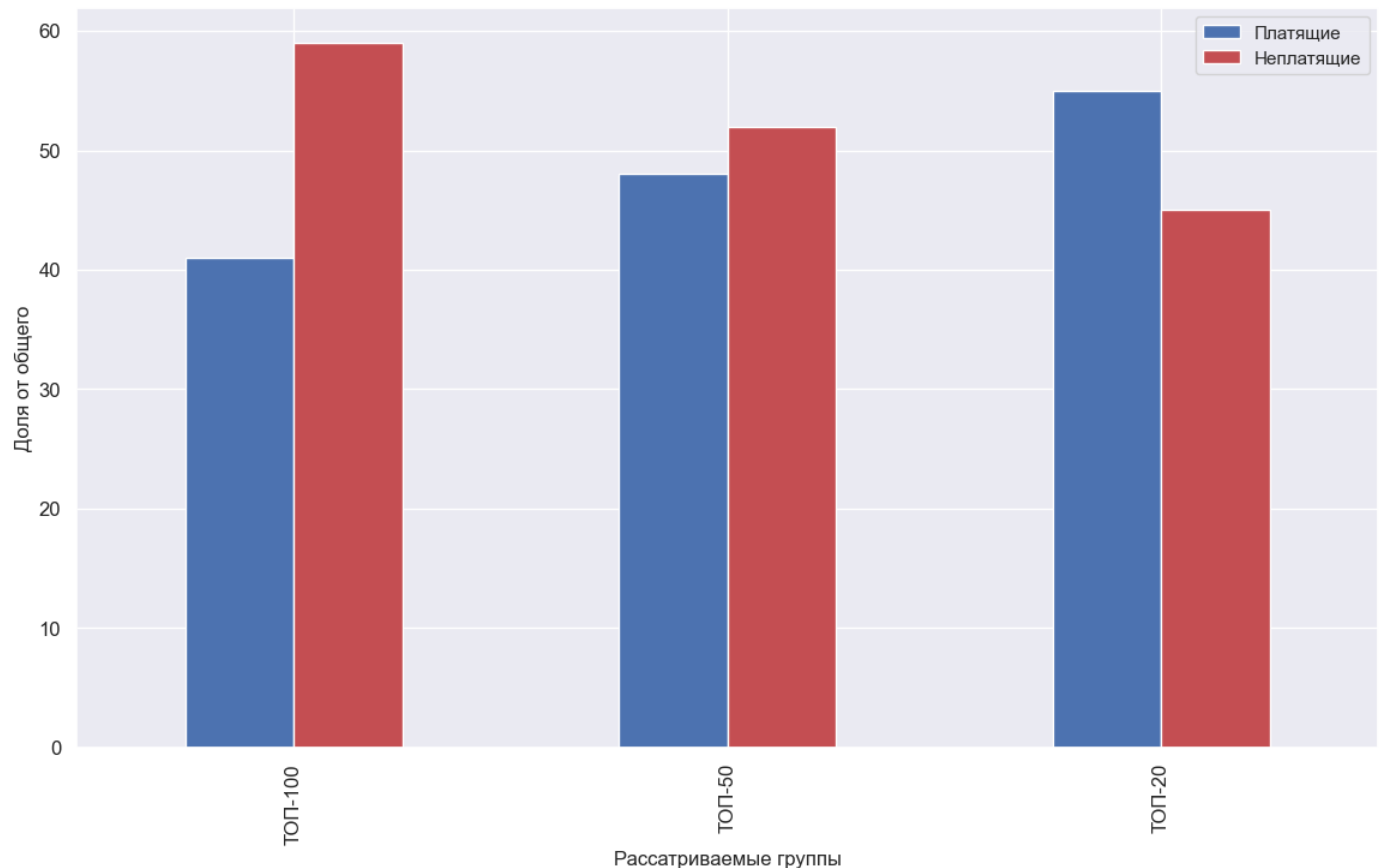
```
In [45]: print('Количество платящих ГМ в топ50: ', f"{len(top100_guilds[top100_guilds.revenue > 0])}")
print('Количество неплатящих ГМ в топ50: ', f"{len(top100_guilds[top100_guilds.revenue == 0])}")
```

Количество платящих ГМ в топ50: 41%  
Количество неплатящих ГМ в топ50: 59%

```
In [46]: print('Количество платящих ГМ в топ50: ', f"{len(top100_guilds.head(50)[top100_guilds.he  
print('Количество неплатящих ГМ в топ50: ', f"{len(top100_guilds.head(50)[top100_guilds.  
  
Количество платящих ГМ в топ50: 48%  
Количество неплатящих ГМ в топ50: 52%
```

```
In [47]: print('Количество платящих ГМ в топ20: ', f"{len(top100_guilds.head(20)[top100_guilds.he  
print('Количество неплатящих ГМ в топ20: ', f"{len(top100_guilds.head(20)[top100_guilds.  
  
Количество платящих ГМ в топ20: 55%  
Количество неплатящих ГМ в топ20: 45%
```

```
In [48]: df = pd.DataFrame({'Рассатриваемые группы': ['ТОП-100', 'ТОП-50', 'ТОП-20'],  
                           'Платящие': [41, 48, 55],  
                           'Неплатящие': [59, 52, 45]})  
df.plot(x='Рассатриваемые группы', y=['Платящие', 'Неплатящие'], kind='bar', color=('b',  
plt.tick_params (axis='both', which='major', labelsize=12)  
plt.ylabel('Доля от общего')  
plt.show()
```



Мы можем наблюдать динамику - чем ближе к верхушке топа, тем больше лидеров топ гильдий предпочитают платить реальные деньги за улучшение своего игрового опыта

**По результатам проведенного исследования, мы не можем опровергнуть нулевую гипотезу о том, что игроки покупающие премиум товары создают более активные гильдии.**

P.S

Несмотря на то, что гипотезу мы опровергнуть не можем, игроки, не покупающие премиум товары, тоже могут похвастаться отличными результатами своих гильдий. Это лишь подтверждает тот факт, что в ММО РПГ взаимодействие игроков является решающим во многом фактором, а механика гильдий дает отличную возможность это взаимодействие реализовать.

## 6. Проверка третьей гипотезы

**Нулевая гипотеза:** Если первое предположение верно, стоит ли существенно повысить стоимость создания гильдии, в том числе сделать это премиум товаром (за реальные деньги).

**Альтернативная гипотеза:** Стоимость создания гильдии повышать не стоит, в том числе делать это премиум товаром.

### 6.1. Делать ли создание гильдии премиум товаром?

Не смотря на то, что первую гипотезу мы опровергли, все-таки стоит рассмотреть несколько факторов, которые могут влиять на число создаваемых неактивных гильдий.

Первым делом еще раз выделим гильдии, которые посчитали неактивными.

```
In [49]: unactive = full_data[(full_data['level'] == 0) &
                             (full_data['total_energy'] < 15000) &
                             (full_data['characters'] == 0)]
unactive.head()
```

```
Out[49]:
```

	datetime	creator_character_id	guild_id	n_added_characters	n_removed_characters	n_talents	level	n_mission
0	2022-12-12 23:13:42	15092203	78698	1	1	0.0	0	
1	2022-12-12 22:29:09	15961764	78604	2	2	0.0	0	
8	2022-12-12 20:06:29	25081103	78416	1	1	0.0	0	
9	2022-12-12 15:29:23	25567139	78228	11	11	0.0	0	
18	2022-12-12 01:23:31	26490083	77993	13	13	0.0	0	

Здесь нас интересует соотношение платящих и неплатящих пользователей, создающих гильдии.

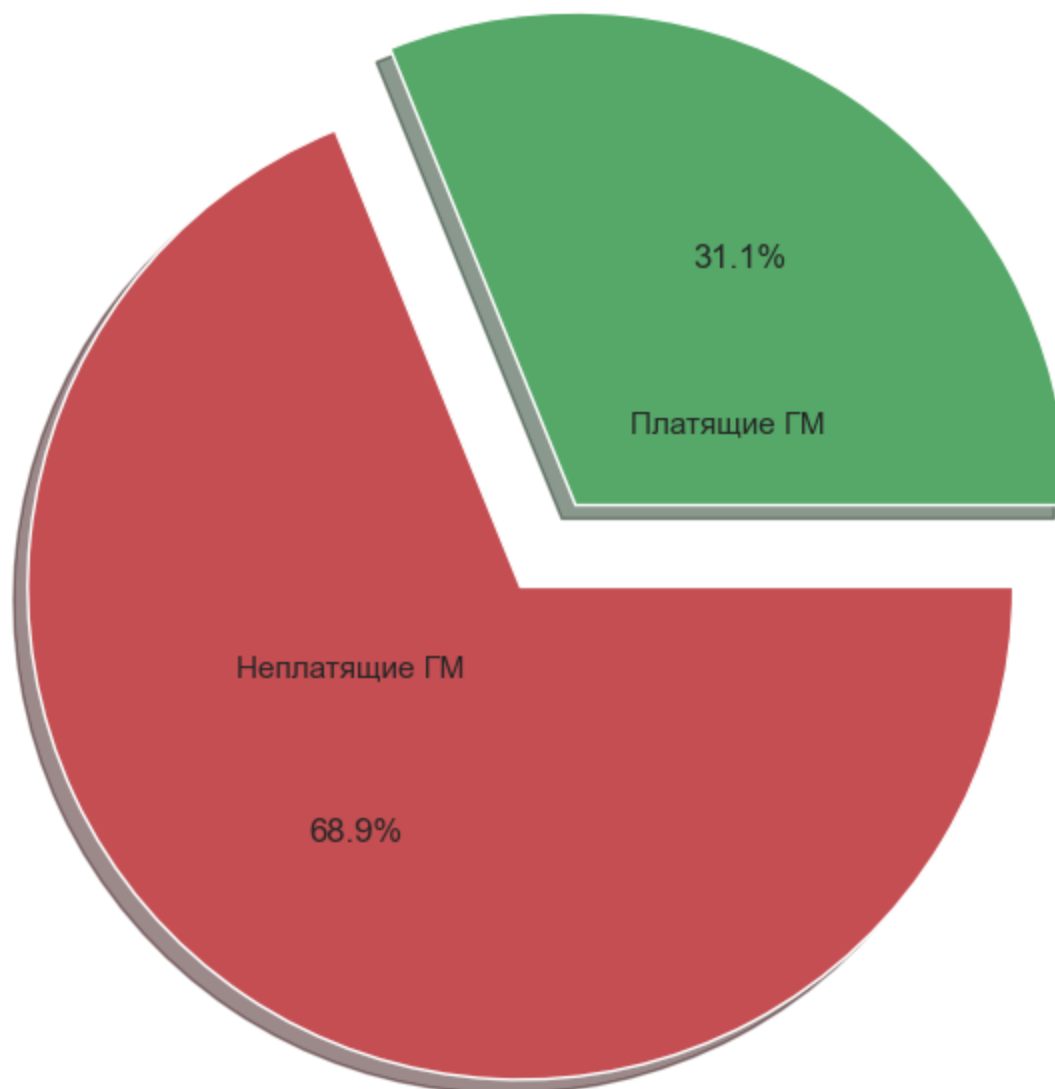
```
In [50]: len(unactive[unactive.revenue > 0]) # платящие
```

```
Out[50]: 52
```

```
In [51]: len(unactive[unactive.revenue == 0]) # не платящие
```

```
Out[51]: 115
```

```
In [52]: plt.pie([52, 115], labels = ['Платящие ГМ', 'Неплатящие ГМ'], colors=['g','r'], autopct='%  
radius = 1, shadow=True, explode=[0.1,0.1], labeldistance = 0.2);
```



**Первое наблюдение:** Большинство неактивных гильдий создаются неплатящими игроками.

Рассмотрим платежеспособность платящих ГМ неактивных гильдий.

```
In [53]: unactive[unactive.revenue > 0]['revenue'].value_counts()
```

```
Out[53]: 0.500000    6  
4.990000    5  
13.680258    2  
293.840000    1  
96.930000    1  
176.860000    1  
38.653619    1  
355.637805    1  
111.980038    1  
36.930000    1  
125.550000    1  
75.829647    1  
34.960000    1  
19.327196    1
```

```
99.560954      1
19.990000      1
11.482006      1
62.737607      1
104.132053     1
42.860000      1
31.359582      1
64.570452      1
59.950000      1
39.950000      1
39.694732      1
15.641257      1
2.571885       1
14.394006      1
17.914776      1
155.697935     1
5.425344       1
66.874075      1
104.970000     1
128.390000     1
24.816398      1
19.970000      1
17.984054      1
39.376720      1
31.843793      1
56.773693      1
32.657964      1
4.440105       1
Name: revenue, dtype: int64
```

```
In [54]: f'{6/52:.0%}'
```

```
Out[54]: '12%'
```

**Второе наблюдение:** Всего 6 из 52 платящих игроков (12%) платили в игру менее 4\$. Большинство платящих игроков вполне платежеспособны.

**Здесь имеет место быть небольшая справка.**

*Как показывают годы наблюдений и исследований поведения игроков касаясь внутриигровых покупок за реальные деньги, если пользователь принес проекту деньги однажды, вероятность совершения повторного платежа становится выше.*

Как можно сопоставить эти знания и результаты наблюдений? Рассмотрим ниже.

## Общее соотношение платящих и неплатящих ГМ.

Очень важно понять - сделав создание ГИ событием за реальные деньги не лишится ли проект большого количества потенциально успешных ГИ?

```
In [55]: # расчет доли платящих/неплатящих ГМ от общего числа

print("Число неплатящих ГМ: ", f"{len(full_data[full_data.revenue == 0]) / 965:.0%}")
print("Число платящих ГМ: ", f"{len(full_data[full_data.revenue > 0]) / 965:.0%}")
```

```
Число неплатящих ГМ:  66%
```

```
Число платящих ГМ:   34%
```

Теперь посмотрим соотношение активных и неактивных гильдий с неплатящими ГМ.



```
In [56]: no_revenue = full_data[full_data.revenue == 0]
len(no_revenue)
```

```
Out[56]: 637
```

```
In [57]: print("Активные гильдии: ", f"{len(full_data[(full_data['level'] > 0) & (full_data['total_revenue'] > 0)])}")
print("Неактивные гильдии: ", f"{len(full_data[(full_data['level'] == 0) & (full_data['total_revenue'] > 0)])}")

Активные гильдии: 72%
Неактивные гильдии: 26%
```

Большинство ГМ активных гильдий никогда не покупали премиум товары.

То есть активность гильдии никак не зависит от того платящий ГМ или нет. В то же время, такое количество неплатящих ГМ может говорить о том что игроки заинтересованы в росте своей ГИ и это является неотъемлемой частью их игрового опыта. Согласны ли будут эти игроки разово отдать некоторую реальную сумму за этот опыт?

**Между нами :) я, из 8 лет в одной ММО РПГ, 6 лет был главой гильдии, которая стала одной из успешнейших. Мне не нужно было вносить деньги, чтобы ее создать и развить. Но заплатил бы я символическую плату за возможность сделать это? Да, потому что в этой механике я вижу главную прелесть ММО RPG. Но знал ли я об этом до создания своей ГИ? Скорее всего - нет.**

## Вывод по пункту 6.1:

**Нельзя однозначно сказать к чему приведет введение платы реальными деньгами за создание гильдии. Некоторые потенциально успешные ГИ могут не появиться из-за "СТОП" на этапе ее создания.**

**В то же время, для заинтересованных в создании гильдии игроков не будет проблемой разово внести минимальный платеж в проект, что повысит конверсию в первый и последующие платежи.**

**Рекомендация:** Для решения данного вопроса может потребоваться больше данных (уровни ГМ на этапе создания ГИ, количество внутриигровой валюты, дата создания аккаунта и т.д.). Также, такое глобальное изменение механики будет правильно рассмотреть с помощью a/b теста чтобы правильно оценить риски нововведения.

## 6.2. Стоит ли существенно повысить стоимость создания гильдии?

Вернемся еще раз к неактивным ГИ.

```
In [58]: unactive.head()
```

```
Out[58]:
```

	datetime	creator_character_id	guild_id	n_added_characters	n_removed_characters	n_talents	level	n_mission
0	2022-12-12 23:13:42	15092203	78698	1	1	0.0	0	
1	2022-12-12 22:29:09	15961764	78604	2	2	0.0	0	
8	2022-12-	25081103	78416	1	1	0.0	0	

	12 20:06:29						
<b>9</b>	2022-12-12 15:29:23	25567139	78228	11	11	0.0	0
<b>18</b>	2022-12-12 01:23:31	26490083	77993	13	13	0.0	0

К сожалению, определяющими факторами для принятия решения были бы данные по уровню персонажа, создающего ГИ и количестве его внутриигровой валюты.

Я проведу анализ экспертно, отталкиваясь от своего опыта.

Для начала, мне интересно взглянуть на количество гильдий, где количество участников никогда не увеличивалось более 3. Вероятнее всего, это гильдии, к примеру:

- новичков, которые пришли в игру и создали гильдию просто так, чтобы бегать под одним тегом. Затем уходит один из них из проекта, а двое переходят в другую, более активную ГИ.
- новичкок встретил механику создания ГИ на 8 уровне персонажа, создал ради интереса, в последствии превратив ГИ в unactive.
- игрок на 8 уровне создает ГИ для возможности выполнять квесты ГИ, бафа, но затем, осознавая полезность активной ГИ, уходит в более крупные

In [59]: 

```
display(unactive[unactive.n_added_characters <= 3])
print("Доля 'временных' ГИ: ", f"{len(unactive[unactive.n_added_characters <= 3]) / 167}:
```

	datetime	creator_character_id	guild_id	n_added_characters	n_removed_characters	n_talents	level	n_missio
<b>0</b>	2022-12-12 23:13:42	15092203	78698	1	1	0.0	0	
<b>1</b>	2022-12-12 22:29:09	15961764	78604	2	2	0.0	0	
<b>8</b>	2022-12-12 20:06:29	25081103	78416	1	1	0.0	0	
<b>26</b>	2022-11-11 18:42:28	17334404	54822	2	2	0.0	0	
<b>27</b>	2022-11-11 15:54:54	19290278	54728	1	1	0.0	0	
...	...	...	...	...	...	...	...	...
<b>934</b>	2022-12-02 14:39:11	24555553	69298	3	3	1.0	0	
<b>936</b>	2022-12-02 13:54:06	24836820	69204	2	2	3.0	0	
<b>944</b>	2022-11-04	17153739	52190	1	1	0.0	0	

18:51:04

948	2022-11-04 20:20:20	20141807	52237	1	1	0.0	0
951	2022-11-04 14:21:02	20697556	51908	1	1	0.0	0

122 rows × 15 columns

Доля 'временных' ГИ: 73%

Как и подозревалось, целых 73% создаваемых ГИ становятся неактивными, скорее всего, по вышеуказанным причинам.

## Что объединяет все эти ГИ?

**Все эти гильдии создаются персонажами низкого уровня, в основном которые способны оплатить создание ГИ собственными средствами из soft-валюты, не оставшись с дырой в кармане.**

Отсюда можно сделать следующий вывод.

## Вывод по пункту 6.2:

**Да, стоит существенно повысить стоимость создания гильдии. Это должно уменьшить количество создаваемых "просто так" гильдий. Также, несколько игроков, заинтересованных в создании общей гильдии, запросто смогут совместно собрать сумму, необходимую на создание ГИ. Таким образом, мы сможем отфильтровать людей заинтересованных в развитии гильдии.**

## По результатам проведенного исследования можно сделать следующие выводы:

Вопрос монетизации в f2p играх всегда стоит очень остро. Но все же, монетизировать можно все, если грамотно это подать, учесть боли игроков и думать о параллельном улучшении игрового опыта игрока.

### 1. Стоит существенно повысить стоимость создания гильдии.

Такое изменение механики приведет к увеличению доли активных ГИ, поможет сделать гонку гильдий более конкурентной, а значит более интересной для игроков. Из-за повышения стоимости создания ГИ, новичкам придется вступать в уже активные гильдии, после чего и сам новичок сможет полноценно оценить игровой опыт в окружении соратников под одним тегом, а сами гильдии будут быстрее расти.

### 2. На данный момент нельзя делать создание гильдии премиум товаром.

Такое изменение, конечно, отсеет всех незаинтересованных в создании гильдии пользователей, но с другой стороны, может "спугнуть" пользователей, которые не привыкли платить в игру реальные деньги.

Для решения данного вопроса может потребоваться больше данных (уровни ГМ на этапе создания ГИ, количество внутриигровой валюты, дата создания аккаунта и т.д.). Также, такое глобальное изменение механики будет правильно рассмотреть с помощью а/в теста чтобы правильно оценить риски нововведения.

**Экспертная оценка:** Введение оплаты реальными деньгами за создание гильдии не является необходимостью. В целях монетизации можно обратить внимание на подмеханики (смена названия ГИ за реальные деньги, смена эмблемы ГИ за реальные деньги, или даже смена главы ГИ за реальные деньги).